

GOVERNMENT OF THE PUNJAB
TECHNICAL EDUCATION & VOCATIONAL TRAINING AUTHORITY



Curriculum For
DAE in
Computer Information Technology
36ADCIT 2024R3
(Revised August- 2024)

CURRICULUM SECTION
ACADEMICS DEPARTMENT

96-H, GULBERG-II, LAHORE

Ph # 042-99263055-9, 99263064

gm.acad@tevta.gop.pk, manager.cur@tevta.gop.pk

**CURRICULUM
OF
DIPLOMA OF ASSOCIATE ENGINEER
IN
COMPUTER INFORMATION
TECHNOLOGY**

(03-YEARS)

36ADCIT 2024R3

Revised (Aug., 2024)

Curriculum Section, TEVTA

Scheme of Studies DAE 1ST Year

Sr. No	Code	Name of Subject	Contact/ Credit (Revised)		
			T	P	C
1.	GEN-111	Islamiat & Pak Studies	1	0	1
2.	TTQ-111/ Civic-111	Tarjumat-ul-Quran / Civics	1	0	1
3.	Eng-112	English	2	0	2
4.	Math-123	Applied Mathematics-I	3	0	3
5.	Phy-132	Applied Physics	1	3	2
6.	Ch-132	Applied Chemistry	1	3	2
7.	CIT-101	Occupational Health Safety & Environment	1	0	1
8.	CIT-103	Introduction to Computing & Application Software	1	6	3
9.	CIT-124	Basic Electronics	2	6	4
10.	CIT-114	Introduction to Computer Programming	2	6	4
11.	CIT-141	Computer & Peripheral Maintenance	0	3	1
Total			15	27	24

Scheme of Studies DAE 2nd Year

Sr. No	Code	Name of Subject	Contact/ Credit (Revised)		
			T	P	C
1	Gen-201	Islamiat & Pak Studies	1	0	1
2	TTQ-211/ Civic-211	Tarjumat-ul-Quran/ Civics	1	0	1
3	Math-233	Applied Mathematics-II	3	0	3
4	Mgm-211	Business Communication	1	0	1
5	CIT-203	Data Communication & Computer Networks	2	3	3
6	CIT-273	Digital Logic Design	2	3	3
7	CIT-222	Operating System	1	3	2
8	CIT-283	Database Management System	2	3	3
9	CIT-293	Computer Graphics Designing	1	6	3
10	CIT-204	Web Development	2	6	4
		Total	16	24	24

Scheme of Studies DAE 3rd Year

Sr. No	Code	Name of Subject	Contact/ Credit (Revised)		
			T	P	C
1.	Gen-301	Islamiat & Pak Studies	1	0	1
2.	CIT-312	Internet of Things	1	3	2
3.	CIT-313	Network and Server Administration	1	6	3
4.	CIT-314	Mobile Applications Development	2	6	4
5.	CIT-322	Entrepreneurship & Freelancing	1	3	2
6.	CIT-343	Microprocessor Fundamentals	2	3	3
7.	CIT-372	UI / UX Design	1	3	2
8.	CIT-381	Software Quality Assurance	0	3	1
9.	CIT-392	Project	0	6	2
Total			9	33	20

سال اول اسلامیات

گھنٹے	ٹی	پی	سی
۱	۰۰	۰۰	۱
۵۰	۵۰	۵۰	۵۰

نشانات

موضوعات

کتاب و سنت

الف: قرآن مجید

- (۱) تعارف قرآن مجید (۲) نزول قرآن (۳) مکی و مدنی سورتوں کی خصوصیات (۴) وحی کی اقسام (۵) پندرہ منتخب آیات مع ترجمہ

- ۱۔ لَنْ تَنَالُوا الْبِرَّ حَتَّى تُنْفِقُوا مِمَّا تَحِبُّونَ
- ۲۔ وَاعْتَصِمُوا بِحَبْلِ اللَّهِ جَمِيعًا وَلَا تَفَرَّقُوا
- ۳۔ وَلَا يَجْزِيكُمْ تَخَنُّنُكُمْ عَلَيَّ أَنْ لَا تَعْلَمُوا
- ۴۔ اِنَّ اللّٰهَ يامرُكُمْ اَنْ تَوَدُّواْ مَا نَحْنُ اِلَيْهِ اَصْلَٰهَآ
- ۵۔ اِنَّ اللّٰهَ يامرُكُمْ بِالْعَدْلِ وَالْاِحْسَانِ
- ۶۔ اِنَّ الصَّلٰوةَ تَنْهٰى عَنِ الْفَحْشَآءِ وَالْمُنْكَرِ
- ۷۔ لَقَدْ كَانَ لَكُمْ فِي رَسُولِ اللّٰهِ اُسْوَةٌ حَسَنَةٌ
- ۸۔ اِنْ اَكْرَمَكُمْ عِنْدَ اللّٰهِ اتَّقَاكُمْ
- ۹۔ وَمَا اتَاكُمْ الرَّسُولُ فَخُذُوهُ وَمَا نَهَىٰ فَعِزُّوهُ
- ۱۰۔ وَاقُفُوْا بَعْدَ اللّٰهِ
- ۱۱۔ وَعَاثِرُوْهُنَّ بِالْمَعْرُوفِ
- ۱۲۔ يٰۤاَيُّهَا اللّٰهُ الرَّبُّ وَاَيُّزِيْ بِيْ اَصْدَقَاتِ
- ۱۳۔ وَاصْبِرْ عَلٰى مَا اَصَابَكَ
- ۱۴۔ وَقُوْلُوْا لَآ اَسَدِيْدَا
- ۱۵۔ اِنَّ الدِّيْنَ عِنْدَ اللّٰهِ السَّلَامُ

سنت

۱۔ سنت کی اہمیت

۲۔ دس منتخب احادیث مع ترجمہ و تشریح

- ۱۔ انما الاعمال بالنیات
- ۲۔ انما بعثتکم مکارم الاخلاق
- ۳۔ لا یؤمن احدکم حتی یتوب الازحیہ ما یحب النفس
- ۴۔ المسلم من سلم المسلمون من لسانہ ویدہ
- ۵۔ قل امنی باللہ ثم استقم
- ۶۔ خیرکم خیرکم لاہلہ
- ۷۔ سباب المسلم فیسوق و قتالہ کفر
- ۸۔ المؤمن اخ المؤمن
- ۹۔ کل المسلم علی المسلم حرام دمه و مالہ و مرضہ
- ۱۰۔ آیتہ المنافی ثلاث اذا حدث کرب و اذا اومن خان و اذا وعد اخلف

۳۔ دین اسلام

- ۱۔ اسلام کے بنیادی عقائد کی وضاحت اور انسان کی انفرادی و اجتماعی زندگی پر ان کے اثرات
- ۲۔ رسالت ۳۔ آخرت ۴۔ ملائکہ ۵۔ آسمانی کتاب
- ۲۔۳۔ عبادات
- ۱۔ نماز ۲۔ روزہ ۳۔ حج ۴۔ زکوٰۃ
- مندرجہ بالا عبادات کی اہمیت و فضیلت، حکمتیں اور انسان کی انفرادی و معاشرتی زندگی پر ان کے اثرات

۴۔ دین اسلام

عمومی مقصد

دین اسلام کے بنیادی عقائد اور عبادات میں جان سکے اور بیان کر سکے

خصوصی مقاصد

- لفظ دین اسلام کے لغوی اور اصطلاحی معنی بیان کر سکے
- دین اسلام کے بنیادی عقائد کی اہمیت بیان کر سکے
- دین اسلام کے بنیادی عقائد کے انسان کی انفرادی و اجتماعی زندگی پر پڑنے والے اثرات بیان کر سکے
- عبادات کے لفظی و اصطلاحی معنی بیان کر سکے
- عقیدے اور عبادات کا فرق بیان کر سکے
- عبادات (نماز، روزہ، حج، زکوٰۃ) کے فوری احکامات اور انسانی زندگی پر ان کے اثرات بیان کر سکے
- اسلامی عقائد اور عبادات کے مطابق اپنی زندگی ڈھال کر ایک اچھا مسلمان بن سکے

سال اول

تدریسی مقاصد

اسلامیات

الف۔ قرآن مجید

عمومی مقصد

طالب علم یہ سمجھنے کے قابل ہو کہ اسلام کی تعلیمات کا اصل سرچشمہ قرآن مجید ہے

خصوصی مقاصد : طالب علم اس قابل ہو جائے گا کہ

- قرآن مجید کی تعریف کر سکے گا

- قرآن مجید کے نزول کی صورت بیان کر سکے گا

- قرآن مجید کی ومدنی سورتوں کی پہچان کر سکے گا

- منتخب آیات کا ترجمہ و تشریح کر سکے گا

عمومی مقصد : طالب علم یہ سمجھنے کے قابل ہو جائے گا کہ منتخب قرآنی آیات کے ذریعے اسلام

کی تعلیمات کا مفہوم کیا ہے

خصوصی مقاصد : طالب علم اس قابل ہو جائے گا کہ

- قرآنی آیات کا ترجمہ و تشریح کر سکے گا

- قرآنی تعلیمات کی روشنی میں اپنی اور معاشرتی اصلاح کر سکے گا

ب۔ سنت

عمومی مقصد : طالب علم نبوی کی اہمیت و ضرورت کو اچھی طرح سمجھنے کے قابل ہو جائے گا

خصوصی مقاصد

- سنت کی تعریف بیان کر سکے گا

- سنت کی اہمیت و ضرورت کی وضاحت کر سکے گا

- سنت کی روشنی میں اسوۂ حسنہ پر عمل کر سکے گا

ج۔ منتخب احادیث نبوی

عمومی مقصد : طالب علم احادیث کی روشنی میں اخلاقی اقدار سے آگاہی حاصل کر سکے گا

خصوصی مقاصد

- احادیث کا ترجمہ و تشریح کر سکے گا

- رسول اللہ ﷺ کے اسوۂ حسنہ کی پیروی کا جذبہ پیدا ہوگا

سال اول

اخلاقیات

(غیر مسلم طلباء کیلئے)

ٹی	پی	سی	گھنٹے
۱	۰۰	۱	نشانات
۵۰	۰۰	۵۰	سال اول

موضوعات

اخلاقیات کی تعریف اور اہمیت
اخلاقیات کا معیار (قانون، عقل، علمی کتب)
مندرجہ ذیل اخلاق کی وضاحت

- دیانت داری
- وفاداری
- نظم و ضبط
- راست گوئی
- صبر و استقلال
- حوصلہ مندی
- وقت کی پابندی
- صفائی
- اعتماد
- باہمی احترام
- مصلحت

سال اول

مطالعہ پاکستان

گھنٹے	ٹی	پی	سی
۱	۰۰	۰۰	۱
نشانات	۵۰	۰۰	۵۰

حصہ دوم

موضوعات

- حریت فکر، مسلمان قوم میں آزادی فکر کی تاریخ، مسلمانوں میں سیاسی آزادی کی اہمیت اور ضرورت، ذہنی و جسمانی غلامی کے نقصانات
- نظریہ پاکستان، قیام پاکستان کی اساس (دین اسلام) قیام پاکستان کی غرض و غایت
- نظریہ پاکستان کی وضاحت، نظریہ پاکستان علامہ اقبال اور قائد اعظم کے ارشادات کی روشنی میں
- نظریہ پاکستان کا تاریخی پہلو، محمد بن قاسم کی آمد
- مجدد الف ثانی اور شاہ ولی اللہ کی تعلیمی خدمات
- سید احمد شہید کی تحریک مجاہدین
- تعلیمی تحریکیں
- علی گڑھ، ندوۃ العلماء، دیوبند، مدرسۃ الاسلام (سندھ)، اسلامیہ کالج (پشاور)، انجمن حمایت اسلام (لاہور)

سال اول تدریسی مقاصد مطالعہ پاکستان

حصہ دوم

تدریسی مقاصد

حریت فکر

عمومی مقصد

طالب علم یہ جان لے کہ اسلام میں اور مسلمان میں آزادی فکر کی کیا اہمیت ہے

خصوصی مقاصد

- حریت فکر کے معنی و مفہوم بیان کر سکے
- آزادی فکر کی اہمیت بیان کر سکے
- خصوصاً اسلام میں آزادی اظہار رائے کی اہمیت بیان کر سکے
- ذہنی غلامی کے قومی سطح پر نقصانات بیان کر سکے
- جسمانی غلامی کے قومی سطح پر نقصانات بیان کر سکے

نظریہ پاکستان

عمومی مقصد

نظریہ پاکستان (دین اسلام) سے پوری طرح واقفیت ہو جائے

خصوصی مقاصد

- نظریہ کی تعریف کر سکے اور اس کی وضاحت کر سکے
- نظریہ پاکستان کی تعریف کر سکے اور اس کا مفہوم بیان کر سکے
- علامہ اقبال اور قائد اعظم کے ارشادات کی روشنی میں نظریہ پاکستان بیان کر سکے

نظریہ پاکستان کا تاریخی پہلو

عمومی مقصد

نظریہ پاکستان کے تاریخی پس منظر سے واقفیت ہو سکے

خصوصی مقاصد

- محمد بن قاسم کے بارے میں بیان کر سکے
- محمد بن قاسم کے ہندوستان پر حملہ کی وجہ بیان کر سکے
- محمد بن قاسم کے ہندوستان پر حملہ کے اثرات بیان کر سکے
- وہ بیان کر سکے کہ ہندوستان میں ہندو مسلم دو قومی نظریہ کا نکتہ آغاز کیا ہے
- مجدد الف ثانی کی علمی خدمات بیان کر سکے
- شاہ ولی اللہ کی علمی خدمات بیان کر سکے
- مجدد الف ثانی اور شاہ ولی اللہ نے جو تبلیغ دین اور مسلمانوں میں سیاسی شعور پیدا کیا اسے بیان کر سکے

علمی تحریک

عمومی مقصد

برصغیر کی علمی تحریکوں سے آگاہی حاصل ہو سکے

خصوصی مقاصد

- علی گڑھ، دیوبند، ندوۃ العلماء، مدرسۃ اسلام، اسلامیہ کالج، انجمن حمایت اسلام نے تعلیم کے ذریعہ سیاسی شعور مسلمانوں میں پیدا کیا اسے بیان کر سکے
- آزادی ہند کے سلسلہ میں تحریک مجاہدین کی خدمات بیان کر سکے

TTQ 111/Civic111 TARJAMA TUL QURAN /
Civics

T	P	C
1	0	1
Total Contact Hours:		
Theory:		32
Practical:		0

As per syllabus of BISE from PCTB Book

Eng-112 English

Total Contact Hours

Theory	64	T	P	C
Practical	0	2	0	2

Learning Outcomes: The students will be able to

Students will be equipped with cognitive skill to enable them to present facts in a systematic and logical manner to meet the language demands of dynamic field of commerce and industry for functional day-to-day use and will inculcate skills of reading, writing and comprehension.

COURSE OUTLINE

Contents	Hours
----------	-------

ENGLISH PAPER "A"

- | | |
|---|---------------|
| 1. PROSE/TEXT | 16 Hrs |
| 1.1. First eight essays of Intermediate English Book-II | |
| 2. CLOZE TEST | 04 Hrs |
| 2.1. A passage comprising 50-100 words will be selected from the text. Every 11th word or any word for that matter will be omitted. The number of missing word will range between 5-10. The chosen word may or may not be the one used in the text, but it should be an appropriate word. | |

ENGLISH PAPER "B"

- | | |
|--|---------------|
| 3. GRAMMAR | 26 Hrs |
| 3.1. Sentence Structure. | |
| 3.2. Tenses. | |
| 3.3. Parts of speech. | |
| 3.4. Punctuation. | |
| 3.5. Change of Narration. | |
| 3.6. One word for several | |
| 3.7. Words often confused | |
| 4. COMPOSITION | 08 Hrs |
| 4.1. Letters/Messages | |
| 4.2. Job application letter | |
| 4.3. For character certificate/for grant of scholarship | |
| 4.4. Telegrams, Cablegrams and Radiograms, Telexes, Facsimiles | |

- 4.5. Essay writing
- 4.6. Technical Education, Science and Our life, Computers, Environmental Pollution, Duties of a Student.

5. TRANSLATION

10 Hrs

- 5.1. Translation from Urdu into English.
- 5.2. For Foreign Students: A paragraph or a dialogue.

RECOMMENDED BOOKS

- 1. Intermediate English Book-II.
- 2. An English Grammar and Composition of Intermediate Level.
- 3. A Hand Book of English Students By Gatherer.

INSTRUCTIONAL OBJECTIVES

PAPER-A

1. DEMONSTRATE BETTER READING, COMPREHENSION AND VOCABULARY

- 1.1 Manipulate, skimming and scanning of the text.
- 1.2 Identify new ideas.
- 1.3 Reproduce facts, characters in own words
- 1.4 Write summary of stories

2. UNDERSTAND FACTS OF THE TEXT

- 2.1 Rewrite words to fill in the blanks recalling the text.
- 2.2 Use own words to fill in the blanks.

PAPER-B

3. APPLY THE RULES OF GRAMMAR IN WRITING AND SPEAKING

- 3.1 Use rules of grammar to construct meaningful sentences containing a subject and a predicate.
- 3.2 State classification of time, i.e. present, past and future and use verb tense correctly in different forms to denote relevant time.
- 3.3 Identify function words and content words.
- 3.4 Use marks of punctuation to make sense clear.
- 3.5 Relate what a person says in direct and indirect forms.
- 3.6 Compose his writings.
- 3.7 Distinguish between confusing words.

4. APPLY THE CONCEPTS OF COMPOSITION WRITING TO PRACTICAL SITUATIONS

- 4.1 Use concept to construct applications for employment, for character certificate, for grant of scholarship.
- 4.2 Define and write telegrams, cablegrams and radiograms, telexes, facsimiles
- 4.3 Describe steps of a good composition writing.
- 4.4 Describe features of a good composition.
- 4.5 Describe methods of composition writing
- 4.6 Use these concepts to organize facts and describe them systematically in practical situation.

5. APPLY RULES OF TRANSLATION

- 5.1 Describe confusion.
- 5.2 Describe rules of translation.
- 5.3 Use rules of translation from Urdu to English in simple paragraph and sentences.

Math-123 Applied Mathematics-I

Total Contact Hours

Theory	96	T	P	C
Practical	0	3	0	3

Learning Outcomes: The students will be able to

- Solve problems of Algebra, Trigonometry, vectors, Mensuration, Matrices and Determinants.
- Develop skill, mathematical attitudes and logical perception in the use of mathematical instruments as required in the technological fields.
- Acquire mathematical clarity and insight in the solution of technical problems.

COURSE OUTLINE

Contents	Hours
1. QUADRATIC EQUATIONS	06 Hrs
1.1. Standard Form Solution	
1.2. Nature of roots	
1.3. Sum & Product of roots	
1.4. Formation	
1.5. Problems	
2. BINOMIAL THEOREM	06 Hrs
2.1. Factorials	
2.2. Binomial Expression	
2.3. Binomial Co-efficient	
2.4. Statement	
2.5. The General Term	
2.6. The Binomial Series	
2.7. Problems.	
3. PARTIAL FRACTIONS	06 Hrs
3.1. Introduction	
3.2. Linear Distinct Factors Case I	
3.3. Linear Repeated Factors Case II	
3.4. Quadratic Distinct Factors Case III	
3.5. Quadratic Repeated Factors Case IV	
3.6. Problems	

- 4. FUNDAMENTALS OF TRIGONOMETRY** **06 Hrs**
- 4.1. Angles
 - 4.2. Quadrants
 - 4.3. Measurements of Angles
 - 4.4. Relation between Sexagesimal & circular system
 - 4.5. Relation between Length of a Circular Arc & the Radian Measure of its central Angle
 - 4.6. Problems
- 5. TRIGONOMETRIC FUNCTIONS AND RATIOS** **06 Hrs**
- 5.1. Trigonometric functions of any angle
 - 5.2. Signs of trigonometric Functions
 - 5.3. Trigonometric Ratios of particular Angles
 - 5.4. Fundamental Identities
 - 5.5. Problems
- 6. GENERAL IDENTITIES** **06 Hrs**
- 6.1. The Fundamental Law
 - 6.2. Deductions
 - 6.3. Sum & Difference Formulae
 - 6.4. Double Angle Identities
 - 6.5. Half Angle Identities
 - 6.6. Conversion of sum or difference to products
 - 6.7. Problems
- 7. SOLUTION OF TRIANGLES** **06 Hrs**
- 7.1. The law of Sines
 - 7.2. The law of Cosines
 - 7.3. Measurement of Heights & Distances
 - 7.4. Problems
- 8. VECTORS AND PHASORS** **12 Hrs**
- 8.1. Scalars and Vectors
 - 8.2. The unit Vectors i, j, k
 - 8.3. Direction Cosines
 - 8.4. Dot Product
 - 8.5. Cross Product
 - 8.6. Analytic Expressions for dot and cross products
 - 8.7. Phasors
 - 8.8. Significance of j Operator
 - 8.9. Different Forms

- 8.10. Algebraic Operations
- 8.11. Problems

9. COMPLEX NUMBERS **09 Hrs**

- 9.1. Introduction and Properties
- 9.2. Basic Operations
- 9.3. Conjugate
- 9.4. Modulus
- 9.5. Different Forms
- 9.6. Problems

10. BOOLEAN ALGEBRA AND GATE NETWORKS **15 Hrs**

- 10.1. Concept and basic laws
- 10.2. Sums of product and products of sums
- 10.3. Binary, decimals and octals, presentation of decimal numbers in BCD
- 10.4. Intercoversion of numbers OR Gates and AND Gates
- 10.5. Logical Expressions and their simplifications
- 10.6. Demorgan's Theorams
- 10.7. NAND Gates and NOR Gates
- 10.8. Problems

11. PLANE ANALYTIC GEOMETRY AND STRAIGHT LINE **06 Hrs**

- 11.1. Coordinate system
- 11.2. Distance formula
- 11.3. Ration Formulas
- 11.4. Inclination and slope of line
- 11.5. Slope Formula
- 11.6. Problems

12. EQUATIONS OF THE STRAIGHT LINE **06 Hrs**

- 12.1. Some Important Forms
- 12.2. General form
- 12.3. Angle Formula
- 12.4. Parallelism and Perpendicularity
- 12.5. Problems

13. EQUATIONS OF THE CIRCLE **06 Hrs**

- 13.1. Standard and Central forms of equations
- 13.2. General Form of Equation
- 13.3. Radius and Coordinates of Center
- 13.4. Problems

RECOMMENDED BOOKS

1. Ghulam Yasin Minhas, Technical Mathematics Vol I, Ilmi Kitab Khana, Lahore
2. Riaz Ali Khan, Polytechnic Mathematics Series Vol- I and Vol -II, Majeed Sons, Faisalabad.
3. Sana Ullah Bhatti, A Text Book of Algebra and Trigonometry, Punjab Text Book Board, Lahore
4. Sana Ullah Bhatti, A Text Book of Analytic Geometry and Calculus, Punjab Text Book Board,

INSTRUCTIONAL OBJECTIVES

1. USE DIFFERENT METHODS FOR THE SOLUTION OF QUADRATIC EQUATION

- 1.1 Define a standard quadratic equation.
- 1.2 Use methods of factorization and method of completing the square for solving the equations.
- 1.3 Derive quadratic formula.
- 1.4 Write expression for the discriminant.
- 1.5 Explain nature of the roots of a quadratic equation.
- 1.6 Calculate the sum and product of the roots.
- 1.7 Form a quadratic equation from the given roots.
- 1.8 Solve problems involving quadratic equations.

2. APPLY BINOMIAL THEOREM FOR THE EXPANSION OF BINOMIAL AND EXTRACTION OF ROOTS.

- 2.1 State binomial theorem for positive integral index.
- 2.2 Explain binomial coefficients:
 $(n,0), (n,1), \dots, (n,r), \dots, (n,n)$
- 2.3 Derive expression for the general term.
- 2.4 Calculate the specified terms.
- 2.5 Expand a binomial of a given index.
- 2.6 Extract the specified roots.
- 2.7 Compute the approximate value to a given decimal place.
- 2.8 Solve problems involving binomials.

3. APPLY DIFFERENT METHODS FOR RESOLVING A SINGLE FRACTION INTO PARTIAL FRACTIONS USING DIFFERENT METHODS

- 3.1 Define a partial fraction, a proper and an improper fraction.
- 3.2 Explain all the four types of partial fractions.
- 3.3 Set up equivalent partial fractions for each type.
- 3.4 Explain the methods for finding constants involved.
- 3.5 Resolve a single fraction into partial fractions.
- 3.6 Solve problems involving all the four types.

4. UNDERSTAND THE SYSTEMS OF MEASUREMENT OF ANGLES.

- 4.1 Define angles and the related terms.
- 4.2 Illustrate the generation of an angle.
- 4.3 Explain sexagesimal and circular systems for the measurement of angles.
- 4.4 Derive the relationship between radian and degree.
- 4.5 Convert radians to degrees and vice versa.
- 4.6 Derive a formula for the circular measure of a central angle.
- 4.7 Use this formula for solving problems.

5. UNDERSTAND BASIC CONCEPTS AND PRINCIPLES OF TRIGONOMETRIC FUNCTIONS.

- 5.1 Define the basic trigonometric functions/ratios of an angle as ratios of the sides of a right triangle.
- 5.2 Derive fundamental identities.
- 5.3 Find trigonometric ratios of particular angles.
- 5.4 Draw the graph of trigonometric functions.
- 5.5 Solve problems involving trigonometric functions.

6. USE TRIGONOMETRIC IDENTITIES IN SOLVING TECHNOLOGICAL PROBLEMS.

- 6.1 List fundamental identities.
- 6.2 Prove the fundamental law.
- 6.3 Deduce important results.
- 6.4 Derive sum and difference formulas.
- 6.5 Establish half angle, double and triple angle formulas.
- 6.6 Convert sum or difference into product and vice versa.
- 6.7 Solve problems.

7. USE CONCEPT, PROPERTIES AND LAWS OF TRIGONOMETRIC FUNCTIONS FOR SOLVING TRIANGLES.

- 7.1 Define angle of elevation and angle of depression.
- 7.2 Prove the law of sines and the law of cosines.
- 7.3 Explain elements of a triangle.
- 7.4 Solve triangles and the problems involving heights and distances.

8. UNDERSTAND PRINCIPLES OF VECTORS AND PHASORS

- 8.1 Define unit vectors i, j, k .
- 8.2 Express a vector in the component form.
- 8.3 Explain magnitude, unit vector, direction cosines of a vector.
- 8.4 Explain dot product and cross product of two vector.
- 8.5 Deduce important results from dot and cross product.
- 8.6 Define phasor and operator j .
- 8.7 Explain different forms of phasors.
- 8.8 Perform basic Algebraic operation on phasors.
- 8.9 Solve problems on phasors.

9. USE PRINCIPLES OF COMPLEX NUMBERS IN SOLVING TECHNOLOGICAL PROBLEMS.

- 9.1 Define a complex number and its conjugate.
- 9.2 State properties of complex numbers.
- 9.3 Give different forms of complex numbers.
- 9.4 Perform basic algebraic operations on complex numbers.
- 9.5 Solve problem involving complex numbers.

10. SOLVE TECHNICAL PROBLEMS USING PRINCIPLES OF BOOLEAN ALGEBRA

- 10.1 Explain fundamental concepts of Boolean algebra
- 10.2 Explain binary numbers, octal numbers, decimal numbers and their inter conversion.
- 10.3 Explain digital addition and multiplication and its applications to OR gates and AND Gates
- 10.4 Illustrate complementation and inversion
- 10.5 Evaluate logical expression
- 10.6 List basic Laws of Boolean Algebra
- 10.7 Explain De-Morgan's theorem
- 10.8 Explain basic duality of Boolean algebra
- 10.9 Derive Boolean expression
- 10.10 Explain combination of GATES
- 10.11 Illustrate sum of products and product of sum
- 10.12 Derive product of sum expression
- 10.13 Explain NAND Gates and NOR Gates
- 10.14 Use the map methods for simplifying expressions
- 10.15 Explain sub-cubes and covering

11. UNDERSTAND THE CONCEPT OF PLANE ANALYTIC GEOMETRY

- 11.1 Explain the rectangular coordinate system.
- 11.2 Locate points in different quadrants.
- 11.3 Derive distance formula.
- 11.4 Describe the ratio formula
- 11.5 Derive slope formula
- 11.6 Solve problems using the above formulae.

12. USE EQUATIONS OF STRAIGHT LINE IN SOLVING PROBLEMS.

- 12.1 Define equation of a straight line.
- 12.2 Derive slope intercept and intercept forms of equations of a straight line.
- 12.3 Write general form of equations of a straight line.
- 12.4 Derive an expression for angle between two straight lines.
- 12.5 Derive conditions of perpendicularity and parallelism of two straight lines.
- 12.6 Solve problems using these equations/formulae.

13. SOLVE TECHNOLOGICAL PROBLEMS USING EQUATIONS OF CIRCLE

- 13.1 Define a circle.
- 13.2 Describe standard, central and general forms of the equation of a circle.
- 13.3 Convert general form to the central form of equation of a circle.
- 13.4 Deduce formula for radius and coordinates of the center of a circle.
- 13.5 Derive equation of the circle passing through three points.
- 13.6 Solve problems involving these equations.

Phy-132 Applied Physics

Total Contact Hours

Theory	32	T	P	C
Practical	96	1	3	2

Learning Outcomes: The students will be able to

The students will be able to understand the fundamental principles and concept of Physics use these to solve problems in practical situations/technological courses and understand concepts to learn advance Physics/technical course.

COURSE OUTLINE

Contents	Hours
1. Measurements	02 Hrs
1.1. Fundamental units and derived units	
1.2. Systems of measurement and S.I. units	
1.3. Concept of dimensions, dimensional formula	
1.4. Conversion from one system to another	
1.5. Significant figures	
2. Scalars and Vectors	03 Hrs
2.1. Revision of head to tail rule	
2.2. Laws of parallelogram, triangle and polygon of forces	
2.3. Resolution of a vectors by rectangular components	
2.4. Multiplication of two vectors, dot product and cross product	
3. Motion	04 Hrs
3.1. Review of laws and equations of motion	
3.2. Law of conservation of momentum	
3.3. Angular motion	
3.4. Relation between linear and angular motion	
3.5. Centripetal acceleration and force	
3.6. Equations of angular motion	
4. Torque, Equilibrium and rotational inertia	04 Hrs
4.1. Torque	
4.2. Centre of gravity and centre of mass	
4.3. Equilibrium and its conditions	
4.4. Torque and angular acceleration	
4.5. Rotational inertia	
5. Sound	05 Hrs
5.1. Longitudinal waves	
5.2. Intensity, loudness, pitch and quality of sound	
5.3. Units of intensity of level and frequency response of ear	
5.4. Interference of sound waves silence zones, beats	

5.5. Acoustics	
5.6. Doppler effect	
6. Light	05 Hrs
6.1. Review laws of reflection and refraction	
6.2. Image formation by mirrors and lenses	
6.3. Optical instruments	
6.4. Waves theory of light	
6.5. Interference, diffraction, polarization of light waves	
6.6. Applications of polarization in sunglasses, optical activity and stress analysis	
7. Optical Fiber	02 Hrs
7.1. Optical communication and problems	
7.2. Review total internal reflection and critical angle	
7.3. Structure of optical fiber	
7.4. Fiber material and manufacture	
7.5. Optical fiber – uses	
8. Lasers	03 Hrs
8.1. Corpuscular theory of light	
8.2. Emission and absorption of light	
8.3. Stimulated absorption and emission of light	
8.4. Laser principles	
8.5. Structure and working of lasers	
8.6. Types of lasers with brief description	
8.7. Applications (Basic concepts)	
8.8. Material processing	
8.9. Laser welding	
8.10. Laser assisted machining	
8.11. Micro machining	
8.12. Drilling, scribing and marking	
8.13. Printing	
8.14. Lasers in medicine	
9. Electromagnetic waves	02 Hrs
9.1. Magnetic field around a current carrying conductor	
9.2. Electric field induced around an changing magnetic flux	
9.3. Moving fields	
9.4. Types of electromagnetic waves	
9.5. Generation of Radio waves	
9.6. Spectrum of electromagnetic waves	
10. Artificial Satellites	02 Hrs
10.1. Review law of gravitation	
10.2. Escape velocity	
10.3. Orbital velocity	
10.4. Geosynchronous and geostationary satellites	
10.5. Use of satellites in data communication	

RECOMMENDED BOOKS

1. Tahir Hussain, Fundamentals of Physics Vol. – I and II
2. FaridKhawaja, Fundamental of Physics Vol. – I and II
3. Wells and Slusher, Schaum’sSeries Physics
4. Nelkon and Oyborn, Advanced Level Practical Physics
5. MehboobIllahi Malik and Inam-ulHaq, Practical physics
6. Wilson, Lasers – Principles and Applications
7. M. Aslam Khan and M. AkramSandhu, Experimental Physics Note Book

INSTRUCTIONAL OBJECTIVES

1. Use concepts of measurement to practical situations and technological problems

- 1.1 Write dimensional formulae for physical quantities.
- 1.2 Derive units using dimensional equations.
- 1.3 Convert a measurement from one system to another.
- 1.4 Use concepts of measurement and significant figures in problem solving.

2. Use concepts of scalars and vectors in solving problems involving these concepts

- 2.1 Explain laws of parallelogram, triangle and polygon of forces.
- 2.2 Describe method of resolution of a vector into components.
- 2.3 Describe method of addition of vectors by rectangular components.
- 2.4 Differentiate between dot product and cross product of vectors.
- 2.5 Use the concepts in solving problems involving addition resolution and multiplication of vectors.

3. Use the law of conservation of momentum and concepts of angular motion to practical situations

- 3.1 Use law of conservation of momentum to practical/technological problem.
- 3.2 Explain relation between linear and angular motion.
- 3.3 Use concepts and equations of angular motion to solve relevant technological problems.

4. Use concepts of torque, equilibrium and rotational inertia to practical situation/problems

- 4.1 Explain Torque.
- 4.2 Distinguish between Centre of gravity and center of mass.
- 4.3 Explain rotational equilibrium and its conditions.
- 4.4 Explain rotational inertia giving examples.
- 4.5 Use the above concepts in solving technological problems.

5. Understand concepts of sound

- 5.1 Describe longitudinal wave and its propagation.
- 5.2 Explain the concepts: Intensity, loudness, pitch and quality of sound.
- 5.3 Explain units of intensity of level and frequency response of ear.
- 5.4 Explain phenomena of silence zones, beats.
- 5.5 Explain acoustics of buildings.
- 5.6 Explain Doppler's Effect giving mathematical expressions.

6. Use the concepts of geometrical optics to mirrors and lenses

- 6.1 Explain laws of reflection and refraction.
- 6.2 Use mirror formula to solve problems.
- 6.3 Use the concepts of image formation by mirrors and lenses to describe working of optical instruments, e.g. microscope, telescopes, camera and sextant.

7. Understand wave theory of light

- 7.1 Explain wave theory of light.
- 7.2 Explain phenomena of interference, diffraction, polarization of light waves.
- 7.3 Describe uses of polarization given in the course contents.

8. Understand the structure, working and uses of optical fiber

- 8.1 Explain the structure of the optical fiber.

- 8.2 Explain its principles of working.
- 8.3 Describe use of optical fiber in industry and medicine.
- 9. Understand the structure, working and uses of lasers**
 - 9.1 Explain the stimulated emission of radiation.
 - 9.2 Explain the laser principle.
 - 9.3 Describe the structure and working of lasers.
 - 9.4 Distinguish between types of laser.
 - 9.5 Describe the applications of lasers in the fields mentioned in the courses contents.
- 10. Understand nature, types, generation and spectrum of electromagnetic waves**
 - 10.1 Explain magnetic field due to current and electric field due to changing magnetic flux.
 - 10.2 Explain moving fields.
 - 10.3 Describe types of electromagnetic waves.
 - 10.4 Explain generation of radio waves
 - 10.5 Explain spectrum of electromagnetic waves.
- 11. Understand types and uses of artificial satellites**
 - 11.1 Explain escape velocity.
 - 11.2 Explain orbital velocity.
 - 11.3 Distinguish between geosynchronous and geostationary satellites.
 - 11.4 Describe uses of artificial satellites in data communication.

List of Practicals: Applied Physics

1. Draw graphs representing the functions:
 - 1.1. $Y=mx$ for $m=0, 0.5, 1, 2$
 - 1.2. $Y=x^2$
 - 1.3. $Y=1/x$
2. Find the volume of a given solid cylinder using Vernier calipers.
3. Find the area of cross-section of the given wire using micrometer screw gauge.
4. Proven that force is directly proportional to (a) mass, (b) acceleration, using Fletcher's trolley.
5. Verify law of parallelogram of forces using Grave-sands apparatus.
6. Verify law of triangle of forces and Lami's theorem.
7. Determine the weight of a given body using:
 - 7.1. Law of parallelogram of forces.
 - 7.2. Law of triangle of forces.
 - 7.3. Lami's theorem.
8. Verify law of polygon of forces using grave-sands apparatus.
9. Locate the position and magnitude of resultant of like parallel forces.
10. Determine the resultant of two unlike parallel forces.
11. Find the weight of a given body using principle of moments.
12. Locate the centre of gravity of regular and irregular shaped bodies.
13. Find Young's Modules of Elasticity of a metallic wire.
14. Verify Hooke's Law using helical spring.
15. Study of frequency of stretched string with length.
16. Study of variation of frequency of stretched string with tension.
17. Study resonance of air column in resonance tube and find velocity of sound.
18. Find the frequency of the given tuning fork using resonance tube.
19. Find velocity of sound in rod by Kundt's tube.
20. Verify rectilinear propagation of light and study shadow formation.
21. Study effect of rotation of plane mirror on reflection.
22. Compare the refractive indices of given glass slabs.
23. Find focal length of concave mirror by locating centre of curvature.
24. Find focal length of concave mirror by object and image method.
25. Find focal length of concave mirror with converging lens.
26. Find refractive index of glass by apparent depth.
27. Find refractive index of glass by spectrometer.
28. Find focal length of converging lens by plan mirror.
29. Find focal length of converging lens by displacement method.
30. Find focal length of diverging lenses using converging lens.
31. Find focal length of diverging lens using concave mirror.
32. Find angular magnification of an astronomical telescope.
33. Find angular magnification of a simple microscope (magnifying glass).
34. Find angular magnification of a compound microscope.
35. Study working and structure of camera.
36. Study working and structure of sextant.
37. Compare the different scales of temperature and verify the conversion formula.

38. Determine the specific heat of lead shots.
39. Find the coefficient of linear expansion of a metallic rod.
40. Find the heat of fusion of ice.
41. Find the heat of vaporization.
42. Determine relative humidity using hygrometer.

Ch-132 Applied Chemistry

Total Contact Hours

Theory	32	T	P	C
Practical	96	1	3	2

Learning Outcomes: The students will be able to

- Understand the significance and role of chemistry in the development of modern technology.
- Becomes acquainted with the basic principles of chemistry as applied in the study of relevant Technology.
- Knows the scientific methods for production, properties and use of materials of industrial & technological significance.
- Gain skill for the efficient conduct of practical's in a chemistry lab.

COURSE OUTLINE

Contents	Hours
1. INTRODUCTION	02 Hrs
1.1 The scope and significance of the subject.	
1.2 Orientation with reference to Technology.	
1.3 Terms used & units of measurements in the study of chemistry.	
2. FUNDAMENTAL CONCEPTS OF CHEMISTRY	02 Hrs
2.1 Symbols, Valency, Radicals, formulas.	
2.2 Chemical Reactions & their types.	
2.3 Balancing of equations by ionic method.	
3. ATOMIC STRUCTURE	02 Hrs
3.1 Sub-atomic particles.	
3.2 Bohr's Atomic Model.	
3.3 The periodic classification of elements and periodic law	
3.4 General characteristics of a period and group.	
4. CHEMICAL BOND	02 Hrs
4.1 Nature of chemical Bond.	
4.2 Electrovalent bond with examples.	
4.3 Covalent Bond (Polar and Non-polar, sigma & Pi Bonds with examples.	
4.4 Co-ordinate Bond with examples.	
5. SOLIDS AND LIQUIDS	03 Hrs
5.1 The liquid and Solids state.	
5.2 The liquids and their general properties (Density, viscosity, surface tension capillary action etc).	

5.3	Solids and their general properties.	
5.4	Crystal structure of solids	
5.5	Crystals of Si and Ge.	
6.	WATER	03 Hrs
6.1	Chemical nature and properties.	
6.2	Impurities.	
6.3	Hardness of water (types, causes & removal)	
6.4	Scales of measuring hardness (Degress Clark, French, PPM, Mgm per litre).	
6.5	Boiler feed water, scales and treatment.	
6.6	Sea-water desalination, sewage treatment.	
7.	ACIDS, BASES AND SALTS	02 Hrs
7.1	Definitions with examples.	
7.2	Properties, their strength, basicity & Acidity.	
7.3	Salts and their classification with examples.	
7.4	pH-value and scale.	
8.	OXIDATION & REDUCTION	02 Hrs
8.1	The process with examples.	
8.2	Oxidizing and reducing agents.	
8.3	Oxides and their classifications.	
9.	NUCLEAR CHEMISTRY	02 Hrs
9.1	Introduction.	
9.2	Radioactivity (Alpha, beta and gamma rays).	
9.3	Half-life process.	
9.4	Nuclear reaction & transformation of elements.	
9.5	Isotopes and their uses.	
10.	ALLOYS	02 Hrs
10.1	Introduction with need.	
10.2	Preparation and properties.	
10.3	Some important alloys and their composition.	
11.	CORROSION	02 Hrs
11.1	Introduction with causes.	
11.2	Types of corrosion.	
11.3	Rusting of Iron	
11.4	Protective measures against corrosion.	
12.	ELECTRO CHEMISTRY	02 Hrs
12.1	Ionization and Arrhenius theory of Ionization.	
12.2	Electrolytes and Electrolysis.	

- 12.3 Faraday's Laws and numerical related to them.
12.4 Application of Electrolysis (Electron, lathing etc).
12.5 Electro Chemical cells.
- 13. ELECTRICAL INSULATING MATERIALS. 02 Hrs**
- 13.1 Introduction.
13.2 Solid insulators with chemical nature.
13.3 Liquid insulators with chemical nature.
13.4 Gaseous insulators with chemical nature.
13.5 Uses and their classification.
- 14. SEMI CONDUCTORS. 02 Hrs**
- 14.1 Introduction
14.2 Atomic structure of silicon and germanium.
14.3 Bonding & Conductivity.
14.4 Energy bands in a semiconductor.
- 15. ETCHING PROCESS. 02 Hrs**
- 15.1 The process and its aims.
15.2 Etching reagents.
15.3 Applications of processors.

RECOMMENDED BOOKS

1. Intermediate Text-Books of chemistry I & II
2. ILMI Applied Science by SH. Ata Mohammed
3. Materials science by J.C.Anderson & Leaver.
4. Polytechnic Chemistry by G.N.Ready (ELBS & Nelson, Hong Kong).
5. Chemistry for engineers by Eric Gyngell.

INSTRUCTIONAL OBJECTIVES

1. UNDERSTAND THE SCOPE, SIGNIFICANCE AND ROLE OF THE SUBJECT.

- 1.1 Define chemistry and its terms.
- 1.2 Define the units of measurements in the study of chemistry.
- 1.3 Explain the importance of chemistry in various fields of specialization.
- 1.4 Explain the role of chemistry in this technology.

2. UNDERSTAND LANGUAGE OF CHEMISTRY AND CHEMICAL REACTIONS.

- 2.1 Define symbol, valence, radical, formula with examples of each.
- 2.2 Write chemical formula of common compounds.
- 2.3 Define chemical reaction and equations.
- 2.4 Describe types of chemical reactions with examples.
- 2.5 Explain the method of balancing the equation by ionic method.

3. UNDERSTAND THE STRUCTURE OF ATOMS AND ARRANGEMENT OF SUB ATOMIC PARTICLES IN THE ARCHITECTURE OF ATOMS.

- 3.1 Define atom.
- 3.2 Describe the fundamental sub atomic particles
- 3.3 Distinguish between atomic no. mass no. and between isotope and isobars.
- 3.4 Explain the arrangements of electrons in different shells and sub energy levels and understand bohr's atomic model.
- 3.5 Explain the grouping and placing of elements in the periodic table especially Si & germanium.
- 3.6 State the periodic law of elements.
- 3.7 Explain the trend of properties of elements based on their position in the periodic table.
- 3.8 Explain general characteristics of a period and a group.

4. UNDERSTAND THE NATURE OF CHEMICAL BONDS.

- 4.1 Define chemical Bond.
- 4.2 State the nature of chemical bond.
- 4.3 Differentiate between electrovalent and covalent bonding.
- 4.4 Explain the formation of polar and non polar, sigma and pi-bond with examples.
- 4.5 Describe the nature of coordinate bond with examples.

5. UNDERSTAND THE STATES OF MATTER AND DISTINGUISHES SOLIDS FROM GASES.

- 5.1 Describe the liquid and solid states of matter.
- 5.2 State the general properties of liquid.
- 5.3 State the general properties of solid.
- 5.4 Explain the formation of crystals and their types.
- 5.5 Describe the crystal structure of Si and Ge.

6. UNDERSTAND THE CHEMICAL NATURE OF WATER.

- 6.1 Describe the chemical nature of water with its formula.
- 6.2 Describe the general impurities present in water.
- 6.3 Explain the causes and methods to remove hardness of water.
- 6.4 Express hardness in different units like mg/litre, p.p.m, degrees Clark and degrees French.
- 6.5 Describe the formation and nature of scales in boiler feed water.
- 6.6 Explain the method for the treatment of scales.
- 6.7 Explain the sewage treatment and desalination of sea water.

7. UNDERSTAND THE NATURE OF ACIDS, BASES AND SALTS.

- 7.1 Define acids, bases and salts with examples.
- 7.2 State general properties of acids and bases.
- 7.3 Differentiate between acidity and basicity.
- 7.4 Define salts, give their classification with examples.
- 7.5 Explain pH value of solution and pH scale.

8. UNDERSTAND THE PROCESS OF OXIDATION AND REDUCTION.

- 8.1 Define oxidation.
- 8.2 Illustrate the oxidation process with examples.
- 8.3 Define reduction.
- 8.4 Explain reduction process with examples.
- 8.5 Define oxidizing and reducing agents and give at least six examples of each.
- 8.6 Define oxides.
- 8.7 Classify the oxides and give examples.

9. UNDERSTAND THE FUNDAMENTALS OF NUCLEAR CHEMISTRY.

- 9.1 Define nuclear chemistry and radio activity.
- 9.2 Differentiate between alpha, beta and gamma particles.
- 9.3 Explain half life process.
- 9.4 Explain at least six nuclear reactions resulting in the transformation of some elements.
- 9.5 State the uses of isotopes.

10. UNDERSTAND THE NATURE OF ALLOYS USED IN THE RESPECTIVE TECHNOLOGY.

- 10.1 Define alloy.
- 10.2 Explain different methods for the preparation of alloys.
- 10.3 Explain important properties of alloys.
- 10.4 Explain the composition, properties and uses of alloys.

11. UNDERSTAND THE PROCESS OF CORROSION.

- 11.1 Define corrosion.
- 11.2 Describe different types of corrosion.

- 11.3 State the causes of corrosion.
- 11.4 Explain the process of rusting of iron.
- 11.5 Describe methods to prevent/control corrosion.

12. UNDERSTAND THE APPLICATION OF ELECTROCHEMISTRY IN DIFFERENT FIELDS OF INDUSTRIES.

- 12.1 Define ionization, electrolyte and electrolysis.
- 12.2 Describe Arrhenius theory of ionization.
- 12.3 State Faraday's laws of electrolysis.
- 12.4 Apply Faraday's laws of different fields of industry.
- 12.5 Solves numerical problem on Faraday's Laws.
- 12.6 Explain the construction and working of Daniel cell and lead accumulator.

13. KNOW THE USE OF INSULATING MATERIALS.

- 13.1 Define insulator, conductor.
- 13.2 Classify solid, liquid and gaseous insulators with their chemical nature.
- 13.3 Describe their uses.

14. UNDERSTAND THE NATURE AND CHEMISTRY OF SEMI CONDUCTORS.

- 14.1 Define semi -conductors.
- 14.2 Draw the atomic structure of silicon and germanium.
- 14.3 Describe the process of bonding and conductivity in conductors and semi-conductors.
- 14.4 Explain energy bands in semi- conductors.

15. USE ETCHING PROCESS IN DIFFERENT FIELDS OF TECHNOLOGY.

- 15.1 Define etching process and its aims.
- 15.2 Enlist the chemicals/reagents used in the process.
- 15.3 Explain the use of the process in the technology.

CIT-101 Occupational Health, Safety & Environment

Total Contact Hours

Theory	32	T	P	C
Practical	0	1	0	1

This introductory course on computer-related health and safety in the IT industry aims to instill a strong safety mindset in students. It covers the various causes of safety hazards in IT workplaces, helping students identify and understand these risks. The course emphasizes the importance of maintaining a safe work environment and teaches theoretical principles of health and safety specific to the IT industry.

Learning Outcomes: The students will be able to

- Recognize Health and Safety Issues in IT: Learn about health and safety problems in IT workplaces and with computers and smart devices.
- Apply Environmental Protection Concepts: Understand and use ideas to protect the environment while working with computers.
- Practice Safety Measures: Follow safety steps to protect physical, mental health of individual and group in computer labs and workplaces.
- Incorporate Health and Safety in Daily Life: Develop habits for maintaining mental and physical health and safety every day, while using computers or working in information technology environment.

COURSE OUTLINE

Contents	Hours
1. Understanding General OHS Concepts	04 Hrs
1.1. What is OHS?	
1.1.1. Introduction to Occupational Health and Safety (OHS)	
1.1.2. OHS regulations and legislation	
1.1.3. Identifying potential workplace hazards and risks	
1.1.4. Roles and responsibilities of employers, employees, and supervisors in maintaining a safe workplace	
2. Physical, Mental, and Social Health Issues	12 Hrs
2.1. Eyesight Issues	
2.1.1. Digital eyestrain, dryness, blurred vision, headaches	
2.1.2. Solutions: 20-20-20 rule, monitor adjustment, avoiding glare, maximizing screen size, screen lighting and setting adjustment, document holder use, artificial tears, washing face with water	

2.2. Wrist, Elbow & Hand, Arm, Shoulder Issues

- 2.2.1. Repetitive strain injury, carpal tunnel syndrome, symptoms of overuse injuries
- 2.2.2. Occupational overuse syndrome, work-related upper limb disorder (WRULD)

2.3. Backbone & Neck Issues

- 2.3.1. Lower back pain or neck pain due to poor workstation design, bad chair, poor posture, sitting for long periods

2.4. Legs and Feet Issues

- 2.4.1. Increased risk of blood clots from sitting without stretching legs

2.5. Hearing Issues

- 2.5.1. Hearing loss from high volume headphone use

2.6. Mental Health Issues

- 2.6.1. Nervous breakdowns, photosensitive epileptic seizures, sleep disorders from late-night device use

2.7. Social Health Issues

- 2.7.1. Stress disorders, behavioral problems, loneliness, depression, anxiety, aggression, low social relationships, obsessive-compulsive disorder (OCD)

2.8. Lifestyle Diseases

- 2.8.1. Obesity, constipation, digestion issues, diabetes, hypertension, cardiovascular issues from long sittings and sedentary lifestyle
- 2.8.2. Solutions: reduce junk and processed diet, soda, caffeine, use green and leafy vegetables, fruits, dairy and protein, eating and sleeping time according to circadian rhythm, daily exercise and social interaction, meditation and prayers

3. Introduction to OHS in Information Technology

04 Hrs

3.1. OHS in IT

- 3.1.1. Understanding OHS regulations and legislation in the IT industry
- 3.1.2. Identifying potential workplace hazards and risks in IT environments

3.2. Ergonomics and Workplace Design

- 3.2.1. Ergonomic principles and best practices in IT workstations
- 3.2.2. Assessing and mitigating ergonomic risks in IT work environments
- 3.2.3. Designing and maintaining a safe and comfortable IT workplace

3.3. Electrical Safety and IT Equipment

- 3.3.1. Electrical hazards in IT environments
- 3.3.2. Safe use and maintenance of IT equipment and tools
- 3.3.3. Proper handling and storage of batteries and other power sources

3.4. Emergencies and IT Incidents

- 3.4.1. Preparing for and responding to emergencies in IT workplaces
- 3.4.2. IT incident response and management
- 3.4.3. IT security and data protection during emergencies

4. OHS Considerations While Working Online

06 Hrs

4.1. Introduction to OHS for Remote Workers

- 4.1.1. Understanding OHS for remote workers
- 4.1.2. Regulations and legislation for remote work
- 4.1.3. Identifying potential workplace hazards and risks while working online

4.2. Ergonomics and Workstation Setup for Remote Work

- 4.2.1. Ergonomic principles and best practices for remote workstations
- 4.2.2. Assessing and mitigating ergonomic risks in remote work environments
- 4.2.3. Designing and maintaining a safe and comfortable remote work environment

4.3. Cyber Security and Information Security for Remote Work

- 4.3.1. Understanding cyber security risks and threats for remote workers
- 4.3.2. Strategies for protecting sensitive personal and organizational information
- 4.3.3. Best practices for secure remote access and online communication
- 4.3.4. Reporting individual cyber security issues to local and online organizations

4.4. Work-Life Balance and Mental Health in Remote Work

- 4.4.1. Maintaining a healthy work-life balance while working remotely
- 4.4.2. Strategies for preventing burnout and promoting mental health and well-being
- 4.4.3. Communicating effectively with colleagues and supervisors while working remotely

4.5. Emergency Preparedness

- 4.5.1. Incident management for remote work
- 4.5.2. Preparing for and responding to emergencies in remote work environments
- 4.5.3. Incident response and management in remote work environments
- 4.5.4. Cyber security and data protection during emergencies while working online

5. Workstation/Workshop Designing 06 Hrs

5.1. Setting Up a Computer Lab

- 5.1.1. Considerations for sunlight, artificial lighting, fresh air, ventilation, social interaction, gender-based harassment, noise, heat, radiation

5.2. Workshop Layout Preparation

- 5.2.1. Designing a suitable workshop layout based on lab space and equipment ratio

5.3. Workstation Setup

- 5.3.1. Minimum working space for individuals and teams

5.4. Routine Cleaning and Maintenance

- 5.4.1. Designing and following SOPs and checklists for cleaning and maintenance

5.5. Precaution Measures

- 5.5.1. Noise, heat, electrostatic discharge precautions, reporting, and correction procedures

5.6. Working Posture and Sitting Position

- 5.6.1. Correct posture and sitting positions concerning table, chair, keyboard, and display

RECOMMENDED BOOKS

1. Safety Practices and Procedures by National Institute of Science and Technical Education (NISTE), Islamabad.
2. "Health and Safety in a Changing World" by Robert Dingwall, Sidney Stark, and Kirstie Ball, 2021, Routledge, ISBN: 9780367350152
3. "Ergonomics for Beginners: A Quick Reference Guide" by Jan Dul and Bernard Weerdmeester, 2019, CRC Press, ISBN: 9780367350138
4. "Digital Minimalism: Choosing a Focused Life in a Noisy World" by Cal Newport, 2019, Publisher: Portfolio, ISBN: 9780525536512

INSTRUCTIONAL OBJECTIVE

1. Understanding General OHS Concepts

- 1.1. Define Occupational Health and Safety (OHS) and its importance.
- 1.2. Describe OHS regulations and legislation.
- 1.3. Identify potential workplace hazards and risks.
- 1.4. Explain the roles and responsibilities of employers, employees, and supervisors in maintaining a safe workplace.

2. Physical, Mental, and Social Health Issues

2.1. Eyesight Issues

- 2.1.1. Explain digital eyestrain, dryness, blurred vision, and headaches.
- 2.1.2. Apply solutions such as the 20-20-20 rule, monitor adjustment, avoiding glare, maximizing screen size, screen lighting and setting adjustment, document holder use, artificial tears, and washing the face with water.

2.2. Wrist, Elbow & Hand, Arm, Shoulder Issues

- 2.2.1. Describe repetitive strain injury, carpal tunnel syndrome, and symptoms of overuse injuries.
- 2.2.2. Identify occupational overuse syndrome and work-related upper limb disorder (WRULD).

2.3. Backbone & Neck Issues

- 2.3.1. Explain lower back pain or neck pain caused by poor workstation design, bad chair, poor posture, and sitting for long periods.

2.4. Legs and Feet Issues

- 2.4.1. Describe the increased risk of blood clots from sitting without stretching legs.

2.5. Hearing Issues

- 2.5.1. Explain hearing loss from high volume headphone use.

2.6. Mental Health Issues

- 2.6.1. Describe nervous breakdowns, photosensitive epileptic seizures, and sleep disorders from late-night device use.

2.7. Social Health Issues

- 2.7.1. Explain stress disorders, behavioral problems, loneliness, depression, anxiety, aggression, low social relationships, and obsessive-compulsive disorder (OCD).

2.8. Lifestyle Diseases

- 2.8.1. Describe obesity, constipation, digestion issues, diabetes, hypertension, and cardiovascular issues from long sittings and sedentary lifestyle.
- 2.8.2. Apply solutions such as reducing junk and processed diet, soda, caffeine, using green and leafy vegetables, fruits, dairy and protein, eating and sleeping time according to circadian rhythm, daily exercise and social interaction, meditation, and prayers.

3. Introduction to OHS in Information Technology

3.1. OHS in IT

- 3.1.1. Understand OHS regulations and legislation in the IT industry.
- 3.1.2. Identify potential workplace hazards and risks in IT environments.

3.2. Ergonomics and Workplace Design

- 3.2.1. Explain ergonomic principles and best practices in IT workstations.
- 3.2.2. Assess and mitigate ergonomic risks in IT work environments.
- 3.2.3. Design and maintain a safe and comfortable IT workplace.

3.3. Electrical Safety and IT Equipment

- 3.3.1. Identify electrical hazards in IT environments.
- 3.3.2. Practice safe use and maintenance of IT equipment and tools.
- 3.3.3. Properly handle and store batteries and other power sources.

3.4. Emergencies and IT Incidents

- 3.4.1. Prepare for and respond to emergencies in IT workplaces.
- 3.4.2. Implement IT incident response and management.
- 3.4.3. Ensure IT security and data protection during emergencies.

4. OHS Considerations While Working Online

4.1. Introduction to OHS for Remote Workers

- 4.1.1. Understand OHS for remote workers.
- 4.1.2. Explain regulations and legislation for remote work.
- 4.1.3. Identify potential workplace hazards and risks while working online.

4.2. Ergonomics and Workstation Setup for Remote Work

- 4.2.1. Apply ergonomic principles and best practices for remote workstations.
- 4.2.2. Assess and mitigate ergonomic risks in remote work environments.
- 4.2.3. Design and maintain a safe and comfortable remote work environment.

4.3. Cyber Security and Information Security for Remote Work

- 4.3.1. Understand cyber security risks and threats for remote workers.
- 4.3.2. Implement strategies for protecting sensitive personal and organizational information.
- 4.3.3. Apply best practices for secure remote access and online communication.
- 4.3.4. Report individual cyber security issues to local and online organizations.

4.4. Work-Life Balance and Mental Health in Remote Work

- 4.4.1. Maintain a healthy work-life balance while working remotely.
- 4.4.2. Implement strategies for preventing burnout and promoting mental health and well-being.
- 4.4.3. Communicate effectively with colleagues and supervisors while working remotely.

4.5. Emergency Preparedness

- 4.5.1. Manage incidents for remote work.
- 4.5.2. Prepare for and respond to emergencies in remote work environments.
- 4.5.3. Implement incident response and management in remote work environments.

- 4.5.4. Ensure cyber security and data protection during emergencies while working online.

5. Workstation/Workshop Designing

5.1. Setting Up a Computer Lab

- 5.1.1. Consider sunlight, artificial lighting, fresh air, ventilation, social interaction, gender-based harassment, noise, heat, and radiation.

5.2. Workshop Layout Preparation

- 5.2.1. Design a suitable workshop layout based on lab space and equipment ratio.

5.3. Workstation Setup

- 5.3.1. Determine minimum working space for individuals and teams.

5.4. Routine Cleaning and Maintenance

- 5.4.1. Design and follow SOPs and checklists for cleaning and maintenance.

5.5. Precaution Measures

- 5.5.1. Implement noise, heat, electrostatic discharge precautions, reporting, and correction procedures.

5.6. Working Posture and Sitting Position

- 5.6.1. Practice correct posture and sitting positions concerning table, chair, keyboard, and display.

CIT-103 Introduction to Computing & Application Software

Total Contact Hours

Theory	32	T	P	C
Practical	192	1	6	3

An application-oriented course on the basic use of computers. It aims to provide the students with hands-on experience of using the computer and its applications. The course covers the immediate application of computers in business and daily use. Students will also learn basic internet, email, and cloud usage.

Learning Outcomes: The students will be able to

- Feel at home while using computers
- Use application software including MS Office, Google Docs, and others
- Use the Internet keeping in mind privacy and safety issues
- Use online services including cloud storage, email, and others

COURSE OUTLINE

Contents	Hours
1. Computers and its Components	07 Hrs
1.1. What is a computer?	
1.2. Working of computer with Block diagram.	
1.3. The System Unit and Motherboard	
1.4. Central Processing Unit (CPU)	
1.5. Memory & Storage Unit.	
1.6. Ports and Expansion Slots	
1.7. Personal Mobile Devices	
2. Input, Output, and Storage	05 Hrs
2.1. Input and Output	
2.2. Data Representation	
2.3. Digital Images	
2.4. Audio & Video	
2.5. Storage Media and Devices	
2.6. Storage File Systems and Management	
2.7. Cloud Storage	
3. Operating Systems	04 Hrs
3.1. Operating System Functions	
3.2. Operating System Interfaces	
3.3. Windows	
3.4. Linux	
3.5. Android	

- 3.6. Macos And IOS
- 4. Application Software 03 Hrs**
- 4.1. What Is Application Software?
 - 4.2. Productivity Software
 - 4.3. Graphics and Multimedia Software
 - 4.4. Communication Software
 - 4.5. Educational Software
 - 4.6. Entertainment Software
 - 4.7. Open-Source Software
 - 4.8. Software Issues and Concerns
- 5. Database Management 03 Hrs**
- 5.1. What Is a Database?
 - 5.2. Database Organization
 - 5.3. Database Management Systems
- 6. The Internet and World Wide Web 04 Hrs**
- 6.1. What Is the Internet?
 - 6.2. How the Internet Works
 - 6.3. Internet Services
 - 6.4. The World Wide Web
 - 6.5. Web Browsing and Searching
 - 6.6. Evaluating online information
 - 6.7. E-Mail, Blogging, and Instant Messaging
 - 6.8. Social Networking
- 7. Communications and Networks 04 Hrs**
- 7.1. What Is a Computer Network
 - 7.2. Benefits of a Computer Network
 - 7.3. Network Fundamentals
 - 7.4. Connecting to the Internet
 - 7.5. Network Communication
 - 7.6. Network Security
- 8. Privacy, Security, and Ethics 02 Hrs**
- 8.1. Understanding common cybersecurity threats
 - 8.2. Protecting personal information online
 - 8.3. Recognizing phishing and scams
 - 8.4. Ethical and Social Issues

RECOMMENDED BOOKS

- Discovering Computers Digital Technology, Data, and Devices The Shelly Cashman Series 2018
- Microsoft Office 6 in 1, Peter Aiken, and others, Que Corporation.
- Teach Yourself Windows in 24 Hrs, Perry G., Sams Publishing.
- MS Office Unleashed, Sams Publishing.

INSTRUCTIONAL OBJECTIVES

1. Computers and its Components

- 1.1. Explain What is a computer
- 1.2. Explain working of computer with Block diagram.
- 1.3. Explain the System Unit and Motherboard
- 1.4. Explain about Central Processing Unit (CPU)
- 1.5. Explain Memory & Storage Unit.
- 1.6. Explain Ports and Expansion Slots
- 1.7. Introduce Personal Mobile Devices

2. Input, Output, and Storage

- 2.1. Explain Input and Output
- 2.2. Explain Data Representation
- 2.3. Explain Digital Images
- 2.4. Explain Audio & Video
- 2.5. Explain Storage Media and Devices
- 2.6. Explain Storage File Systems and Management
- 2.7. Explain Cloud Storage

3. Operating Systems

- 3.1. Introduce Operating System Functions
- 3.2. Introduce Operating System Interfaces
- 3.3. Introduce Windows
- 3.4. Introduce Linux
- 3.5. Introduce Android
- 3.6. Introduce MacOS and iOS

4. Application Software

- 4.1. Explain What Is Application Software?
- 4.2. Explain Productivity Software
- 4.3. Explain Graphics and Multimedia Software
- 4.4. Explain Communication Software
- 4.5. Explain Educational Software
- 4.6. Explain Entertainment Software
- 4.7. Explain Open-Source Software
- 4.8. Explain Software Issues and Concerns

5. Database Management

- 5.1. Explain What Is a Database?
- 5.2. Explain Database Organization
- 5.3. Explain Database Management Systems

6. The Internet and World Wide Web

- 6.1. Explain What Is the Internet?
- 6.2. Explain How the Internet Works
- 6.3. Explain Internet Services

- 6.4. Explain The World Wide Web
- 6.5. Explain Web Browsing and Searching
- 6.6. Explain Evaluating online information
- 6.7. Explain E-Mail, Blogging, and Instant Messaging
- 6.8. Explain Social Networking

7. Communications and Networks

- 7.1. Explain What Is a Computer Network
- 7.2. Explain Benefits of a Computer Network
- 7.3. Explain Network Fundamentals
- 7.4. Explain Connecting to the Internet
- 7.5. Explain Network Communication
- 7.6. Explain Network Security

8. Privacy, Security, and Ethics

- 8.1. Explain Understanding common cybersecurity threats
- 8.2. Explain Protecting personal information online
- 8.3. Explain Recognizing phishing and scams
- 8.4. Explain Ethical and Social Issues

List of Practical's

1. Introduction To Computer: End-User Point Of View

- 1.1. Basic Design of Computer
- 1.2. Hierarchy of Memory
- 1.3. External View of Computer
- 1.4. Input and Output Devices
- 1.5. Computer Software

2. Windows Operating System

- 2.1. Computer Bootstrap
- 2.2. Install operating system
- 2.3. Install peripheral devices
- 2.4. Install software application
- 2.5. Update/upgrade software application
- 2.6. Using and Customizing Desktop
- 2.7. File Management
- 2.8. Customizing Control Panel
- 2.9. Using scanning
- 2.10. Printer setup

3. Typing Master

- 3.1. Learning Keyboard
- 3.2. Skill improvement Paragraphs
- 3.3. Speed Paragraphs
- 3.4. Rules for typing Manuscript or Reports

4. MS Office (Word Processing)

- 4.1. Create and Manage Documents
- 4.2. Format Text, Paragraphs, and Sections
- 4.3. Create Tables and Lists
- 4.4. Apply References
- 4.5. Insert and Format Objects
- 4.6. Mail Merge
- 4.7. Document Views and Printing Documents

5. Urdu Word Processing (In-Page or MS Word)

- 5.1. Set keyboard preferences
- 5.2. Layout Page
- 5.3. Toggle between Languages
- 5.4. Insert Columns
- 5.5. Shortcuts

6. MS Office (Excel)

- 6.1. Create and manage worksheets and workbooks
- 6.2. Data Types

- 6.3. Entering and Editing Data
- 6.4. Data Manipulation
- 6.5. Create tables
- 6.6. Apply formulas and functions
- 6.7. Create charts and objects
- 6.8. Formatting Cells
- 6.9. Printing Worksheet

7. MS Office (PowerPoint)

- 7.1. Introduction to PowerPoint & its interface
- 7.2. Insert Slide
- 7.3. Design Slide
- 7.4. Entering and Editing Presentation Objects
- 7.5. Apply Transitions and Animations
- 7.6. Presentation Views
- 7.7. Themes
- 7.8. Using Printing Options

8. MS Office (MS Access)

- 8.1. Create and manage a database
- 8.2. Creating Database Tables
- 8.3. Setting Up Database Tables
- 8.4. Creating Queries for Tables
- 8.5. Creating Data Entry Forms for Tables

9. Internet, Cloud Storage & Electronic Mail

- 9.1. Using different search engines
- 9.2. Create and operate a Gmail account
- 9.3. Access Google Drive using a browser
- 9.4. Create a Google Doc
- 9.5. Create a Google Sheet
- 9.6. Create a Google Slide
- 9.7. Design a Google form to collect information about students
- 9.8. Search the Internet for fake news during the last 3 months and prepare a report

10. Graphics And Image

- 10.1. How to scan images using scanning software
- 10.2. Editing Picture
- 10.3. Saving Graphics

11. Video

- 11.1. Understanding Video software
- 11.2. How to Play Video (Any Format)

CIT- 124 Basic Electronics

Total Contact Hours

Theory	64	T	P	C
Practical	192	2	6	4

In our course, you will gain a thorough understanding of both qualitative and quantitative concepts in electromagnetism, linear circuits, and basic electronics. We will explore the operating principles behind various electrical and electronic devices and learn how to apply these principles to solve practical technological problems. You will also see how electronic and electrical concepts integrate with other fields of study. Our focus will include fundamental concepts and applications in analogue electronics, particularly how transistors are used for amplification. By the end of the course, you should be able to use your knowledge to effectively maintain and troubleshoot electronic devices.

Learning Outcomes: The students will be able to

- Understand qualitatively and quantitatively basic concepts and principles of electromagnetism,
- Linear circuits and basic electronics.
- State and explain the operating principles applied to electrical and electronic devices.
- Apply the Concepts and principles to solve related practical technological problems.
- Relate electronic and electrical concepts and principles to other discipline of study
- Understand basic concepts, principles and application of analogue electronics
- Apply operational principles of transistors for amplification.
- Use knowledge of electronics to maintain electronic devices

COURSE OUTLINE

Contents	Hours
1. Basic Principles of Electricity	12 Hrs.
1.1. Fundamentals of Electricity	
1.2. Understand electron theory	
1.3. Basic Circuit Concepts	
1.4. Electromagnetism	
2. Magnetism, Electromagnetism & Electrostatics.	10 Hrs
2.1. Understand Magnetism	
2.2. Understand Electromagnetism	
2.3. Understand Electrostatics	
3. DC Fundamentals.	09 Hrs
3.1. Understand Ohm's Law	
3.2. Understand Laws of Resistance	

- 3.3. Understand Kirchhoff's Laws
- 3.4. Understand Resistors
- 3.5. Understand Batteries
- 4. AC Fundamentals. 09 Hrs**
- 4.1. The simple AC generator.
- 4.2. AC Circuits
- 5. Semiconductor Diodes, 08 Hrs**
- 5.1. Understand principles, characteristics and application of various types of Semiconductor diodes.
- 5.2. PN Junction Theory:
- 5.3. Understand PN Diode Applications
- 6. Understand Resonance, Filters & Coupling Circuits 08 Hrs**
- 6.1. Understand Resonance
- 6.2. Understand Filters
- 6.3. Understand Coupling Circuits
- 7. Understand Transistors. Operational Amplifiers and Special Devices 08 Hrs**
- 7.1. Transistors
- 7.2. Operational Amplifiers
- 7.3. Special Devices
- 7.4. Define Silicon Controlled Rectifier (SCR) and Simple SCR Applications.
- 7.5. Define Diac and Triac
- 7.6. Define Unijunction Transistor (UJT)
- 7.7. Define Photo diode & Phototransistor
- 7.8. Define Light Activated SCR (LASCR)
- 7.9. Define Opto-coupler.

RECOMMENDED BOOKS

1. Bird J O — Electrical and Electronic Principles and Technology, Second Edition (Newnes, 2004) ISBN 0750665505
2. Bird J O — Electrical Circuit Theory and Technology (Newnes, 2004) ISBN 0750657847
3. Grob, Bernard, Basic Electronics, Eight Edition.
4. TL Floyd "Electronics Devices" 8th ed. Prentice Hall, ISBN 0131140809
5. Floyd "Digital Fundamentals"

INSTRUCTIONAL OBJECTIVES

1. Basic Principles of Electricity

- 1.1. Fundamentals of Electricity and Understand Electrical Quantities
 - 1.1.1. Define the electric charge (Q), Current (I), AC and DC Current, Source, Load, Path, Electricity, Voltage.
 - 1.1.2. Why Does Current Flow?
 - 1.1.3. Describe Ohm's Law and the Ohms Law importance
 - 1.1.4. Characteristics of Ohm's Law
 - 1.1.5. Define Passive Devices, Active Devices
 - 1.1.6. Define Power (Watts), Conservation of Power, Relation between Voltage, Power, and Current
 - 1.1.7. Define Resistance, Resistors, Resistor Color Code, Conductance (G), Resistivity (ρ), and Resistivity of Materials
 - 1.1.8. Define Conductors, Semiconductors, Insulators
 - 1.1.9. Power used by Resistors ($P = I^2 \cdot R$)
- 1.2. Understand electron theory.
 - 1.2.1. Describe the structure of atom.
 - 1.2.2. Describe the K, L, and M shells.
 - 1.2.3. Describe energy level.
 - 1.2.4. Describe valence electron.
 - 1.2.5. Energy bands with reference to conductors, insulators & semiconductors
- 1.3. Basic Circuit Concepts
 - 1.3.1. Define a Circuit
 - 1.3.2. Describe Circuit Diagram (Series circuit, Parallel circuit, Combination of Series and Parallel Circuits)
 - 1.3.3. Parallel Circuits)
 - 1.3.4. Polarity and direction of current flow

2. Magnetism, Electromagnetism & Electrostatics.

2.1. Understand Magnetism

- 2.1.1. Define lines of force, flux, flux density, permeability, reactance & their unit's
- 2.1.2. Explain properties of magnetic lines of force
- 2.1.3. Describe types of magnets
- 2.1.4. Explain magnetic properties of materials
- 2.1.5. Define and list ferromagnetic, paramagnetic and diamagnetic materials.
- 2.1.6. Describe magnetic induction.

2.2. Understand Electromagnetism

- 2.2.1. Define electromagnetism, magneto-motive force, field intensity ($H=AT/L$) .
- 2.2.2. Describe magnetic hysteresis
- 2.2.3. Explain electromagnetic induction
- 2.2.4. Explain magnetic field around a current carrying conductor
- 2.2.5. Define inductor
- 2.2.6. Explain force between two magnetic fields and motor action
- 2.2.7. Define Faraday's law of electromagnetic induction ($e = N d\Phi/dt$)

2.2.8. Define Lenz's Law

2.3. Understand Electrostatics.

2.3.1. Describe principle of electrostatic charges

2.3.2. Explain the effect of negative & positive charges

2.3.3. Describe the laws of electrostatics

2.3.4. Describe electrostatic induction & field strength

2.3.5. Explain properties of electric lines of force

2.3.6. Comparison between electric lines of force and magnetic lines of force

2.3.7. Describe dielectric & dielectric strength/dielectric constant

2.3.8. Describe the importance of dielectric & dielectric strength

2.3.9. Describe capacitor and capacitance

2.3.10. Define breakdown voltage

2.3.11. Describe the capacitance of parallel plate capacitor

2.3.12. Describe types of capacitors and the use/applications of capacitors

2.3.13. Calculate the total capacitance in series in parallel and series-parallel combination

2.3.14. Explain the energy stored in capacitor

2.3.15. Describe the color coding, tolerance and voltage rating of capacitors

2.3.16. Describe the troubles/problems use of capacitors

3. DC FUNDAMENTALS.

3.1. Understand Ohm's Law.

3.1.1. Define ohm's law

3.1.2. Solve problems using Ohm's law

3.2. Understand Laws of Resistance

3.2.1. Define specific resistance

3.2.2. Define conductor and conductivity

3.2.3. Explain the effect of temperature on resistance

3.2.4. Define coefficient of resistance

3.2.5. Solve problems using $R = \rho \times L/A$ and $R_t = R_o (1 + \alpha t)$.

3.2.6. Describe the resistance in series, parallel and in series-parallel Circuits

3.2.7. Calculate the combination of resistances and cells, R_t , I & V .

3.2.8. Define power and energy

3.2.9. Define units of power and energy

3.2.10. Explain the power dissipation in resistors

3.3. Understand Kirchhoff's' Laws

3.3.1. Define Kirchhoff's laws

3.3.2. Solve problems using Kirchhoff voltage law

3.3.3. Solve problems using Kirchhoff current law

3.4. Understand Resistors

3.4.1. Define resistance and resistor

3.4.2. Describe types of resistors

3.4.3. Enlist use of resistors

3.4.4. Describe resistor color codes

3.4.5. Describe power rating of resistor

3.5. Understand Batteries

- 3.5.1. Name types of D.C source
- 3.5.2. Describe types of cells (Mercury, Silver oxide, Nickel cadmium)
- 3.5.3. Describe lead acid battery
- 3.5.4. Describe solar cells
- 3.5.5. Explain the internal resistance of cell
- 3.5.6. Application of cell as constant voltage and constant current source.

4. AC Fundamentals.

4.1. The simple AC generator.

- 4.1.1. Define Sine wave, cycle, wavelength, period, frequency and units.
- 4.1.2. Define AC sine wave form and define its characteristics. (Instantaneous, peak, average, RMS).
- 4.1.3. Define Audio and Radio frequencies, wavelengths and periods frequency spectrum.
- 4.1.4. Describe Types of alternating waveforms (sinusoidal and non-sinusoidal waves).

4.2. AC Circuits

- 4.2.1. Describe AC through pure resistance and its Phasor quantities (Phase angle, in-phase, out of phase waves and phase lag & lead and power factor).
- 4.2.2. Calculation of V, I and W for resistive circuits through inductance.
- 4.2.3. Define Self-inductance and self-induced voltage.
- 4.2.4. Explain Inductive reactance ($X_L = 2\pi fL$), Phase relation between V & I.
- 4.2.5. Explain Phasor diagram and power for pure inductor.
- 4.2.6. Define AC through R-L series circuit.
- 4.2.7. Define power in a R-L series circuit.
- 4.2.8. Time constant, $\tau = L/R$, and its effect.
- 4.2.9. Describe Impedance, Impedance triangle.
- 4.2.10. Explain AC through R-L parallel circuit.
- 4.2.11. Define Inductive reactance in series, parallel and series-parallel combination.
- 4.2.12. Explain AC through pure capacitor, Phase relation between V & I and power.
- 4.2.13. Define Capacitive Reactance
- 4.2.14. Define AC through R-C series circuit.
- 4.2.15. Define Time constant RC and its effect.
- 4.2.16. Explain AC through R-C parallel circuit.
- 4.2.17. Capacitive Reactance in series, parallel, and series parallel combination.
- 4.2.18. AC through RLC series circuit, phase relation and power calculation.
- 4.2.19. AC through RLC parallel circuit phase relation and power calculation.

5. Semiconductor Diodes

5.1. Understand principles, characteristics and application of various types of Semiconductor diodes.

- 5.1.1. Explain semiconductor doping
- 5.1.2. List donor and acceptor materials for silicon & germanium
- 5.1.3. Define majority carries and minority charge carriers.
- 5.1.4. Explain the effect of temperature & light on the resistance of

- 5.1.4.1. Intrinsic semiconductor
- 5.1.4.2. Extrinsic semiconductor

5.2. PN Junction Theory:

- 5.2.1. Draw a PN Junction
- 5.2.2. Define the terms depletion layer capacitance & diffusion capacitance.
- 5.2.3. Sketch the voltage-current characteristics curve for a PN junction.
- 5.2.4. Describe R_F , R_R and I_s from the diode characteristics curve.
- 5.2.5. List the typical values of barrier potentials for silicon and germanium diode.

5.3. Understand PN Diode Applications

- 5.3.1. List the uses of PN diode.
- 5.3.2. Explain half and full wave rectifier using circuit diagram.
- 5.3.3. Define Ripple factor, surge current.
- 5.3.4. Explain function of rectifier (L, PI, T) filters
- 5.3.5. Explain its uses as voltage multiplier (doubler).
- 5.3.6. Explain the working of a voltage doubler circuit.
- 5.3.7. List the applications of voltage multiplier circuit.
- 5.3.8. Explain the operation of a diode as a switch.

6. Understand Resonance, Filters & Coupling Circuits

6.1. Understand Resonance

- 6.1.1. Explain resonance
- 6.1.2. Explain the relation between frequency, inductance & capacitance at resonant
- 6.1.3. Draw the series resonant circuit
- 6.1.4. Explain series resonant circuit
- 6.1.5. Calculate current, voltage and impedance of series resonant circuit
- 6.1.6. Calculate current, voltage and impedance of series resonant circuit
- 6.1.7. Draw the diagram of parallel resonant circuit
- 6.1.8. Explain the parallel resonant circuit
- 6.1.9. Compare series and parallel resonant circuit
- 6.1.10. Describe the bandwidth of a resonant circuit
- 6.1.11. Enlist the use of resonant circuit.

6.2. Understand Filters

- 6.2.1. Explain purpose & action of a filter circuit
- 6.2.2. Enlist the types of filter circuits
- 6.2.3. Explain low pass filter high pass filter, K-filter & m-derived filters
- 6.2.4. Explain band pass filter & band stop filter

6.3. Understand Coupling Circuits

- 6.3.1. Explain action & purpose of a coupling circuit

7. Understand Transistors. Operational Amplifiers and Special Devices

7.1. Transistors

- 7.1.1. Define Transistor and its types
- 7.1.2. Describe Basic Transistor operation, Forward, Reverse Bias. Transistor current.
- 7.1.3. Transistor Parameters and Ratings

- 7.1.4. Transistor as a voltage amplifier.
- 7.1.5. Transistor, modes of operation.
- 7.1.6. Transistor as a switch.
- 7.1.7. Transistor Clipper
- 7.1.8. Describe Junction Field Effect Transistor (JFET).
- 7.1.9. Describe Metal oxide Semiconductor FET (MOSFET) types.

7.2. Operational Amplifiers

- 7.2.1. Define Operational Amplifier (OP-Amp)
- 7.2.2. Define the types of (OP-Amp)
 - 7.2.2.1. The Non-inverting Voltage Amplifier
 - 7.2.2.2. Inverting Voltage Amplifier
- 7.2.3. Inverting Current-to-Voltage Transducer
- 7.2.4. Non-inverting Voltage-to-Current Transducer
- 7.2.5. Inverting Current Amplifier
- 7.2.6. Summing Amplifiers
- 7.2.7. Non-Inverting Summing Amplifier
- 7.2.8. Differential Amplifier
- 7.2.9. (OP-Amp) as Adder/Subtractor

7.3. Special Devices

- 7.3.1. Define Zener Diodes.
 - 7.3.1.1. Zener Diode as voltage Regulator, percentage of regulation.
 - 7.3.1.2. Zener limiting.
- 7.3.2. Define Optical Diodes
 - 7.3.2.1. Light Emitting Diode (LED)
 - 7.3.2.2. Liquid crystal Diode (LCD)
 - 7.3.2.3. Photo diode.
 - 7.3.2.4. LASER Diode.
- 7.3.3. Define Silicon Controlled Rectifier (SCR) and Simple SCR Applications.
- 7.3.4. Define Diac and Triac
- 7.3.5. Define Unijunction Transistor (UJT)
- 7.3.6. Define Photo diode & Phototransistor
- 7.3.7. Define Light Activated SCR (LASCR)
- 7.3.8. Define Opto-coupler.

LIST OF PRACTICAL

1. Study of Ammeter, Voltmeter and Multimeter.
2. Measurement of current, voltage and resistance.
3. Verification of Ohm's Law by:
 - 3.1. Keeping the voltage constant.
 - 3.2. Keeping the resistance constant.
4. Verify the laws of series and parallel combination of resistances by
 - 4.1. Ohmmeter method.
 - 4.2. Voltmeter-Ammeter method.
5. Determine temperature coefficient of resistance.
6. Verify Kirchhoff's laws.
7. Measurement of power by:
 - 7.1. Voltmeter/Ammeter method.
 - 7.2. Wattmeter.
8. Measurement of Energy by:
 - 8.1. Wattmeter and clock method.
 - 8.2. Energy meter.
9. Practice of resistor color-coding.
10. Combine cells in series and parallel and verify the net voltage.
11. Study of lead acid battery, practice and use of hydrometer.
12. Practice charging of a lead acid battery.
13. Study of various types of capacitors and their color-coding.
 - 13.1. Verify laws of combination of capacitors.
 - 13.2. Observe capacitor charging and discharging.
14. Study of sine wave on an oscilloscope and determine its peak, peak-to-peak, R.M.S. and average values of current and voltage.
15. Determine of wavelength, time- period and frequency of a given AC signal by oscilloscope.
16. Study the behavior of inductance and capacitance with AC and DC supplies.
17. Practice of series parallel resistor circuits and verification of OHM's LAW.
18. Study and perform forward and reverse biasing of diodes
 - 18.1. Study the characteristics of forward and reverse biased junction diodes.
 - 18.2. Measure voltage across the diode when resistor is in series with diode.
19. Soldering (Use of Basic Tools, Use of Soldering and De-soldering Tools, Soldering Techniques)
20. PCB Design using soldering techniques follow the listed below steps
21. Selection of PCB
22. Cutting Sheets
23. Transfer of Circuit
24. Chemical Etching Process
25. Identification and Codes
26. Mounting and Soldering Components
27. PCB Tests

CIT-114 Introduction to Computer Programming

Total Contact Hours

Theory	64	T	P	C
Practical	192	2	6	4

A first course in programming. The first programming language is introduced and fundamental programming skills are developed in the students during the course. This course also introduces basic object-oriented concepts and allows students to practice them for effective software development. The course is supported by a significant laboratory component to complement classroom discussions.

Learning Outcomes: The students will be able to

- Understand fundamentals of programming with C++
- Use C++ programming language to solve various kinds of problems
- Apply object-oriented programming for designing solutions

COURSE OUTLINE

Contents	Hours
1. Introduction to Programming	04 Hrs
1.1. Giving instructions to humans vs machines	
1.2. Programming Computers	
1.3. Brief History of Programming	
1.4. Programming Languages	
1.5. Categories of Programming Languages	
1.6. Features of Programming Languages	
2. C++ Fundamentals	06 Hrs
2.1. Structure of a C++ program	
2.2. Introducing operators	
2.3. Reading from keyboard	
2.4. Introducing control statements	
2.5. The concept of code blocks	
2.6. Introducing functions	
3. Data Types and Operators	04 Hrs
3.1. C++ Data Types	
3.2. Literals	
3.3. Variables	
3.4. Arithmetic Operators	
3.5. Relational and Logical Operators	
3.6. The Assignment Operator	

3.7. Compound Statements	
3.8. Type Conversion in Assignments	
3.9. Type Conversion in Expressions	
3.10. Casts	
3.11. Spacing and Parentheses	
4. Program Control Statements	08 Hrs
4.1. The if Statement	
4.2. The if-else Statement	
4.3. The else-if Statement	
4.4. The switch Statement	
4.5. The for Loop	
4.6. The while Loop	
4.7. The do while Loop	
4.8. Nested Loop	
5. Arrays and Strings	08 Hrs
5.1. One-Dimensional Arrays	
5.2. Two-Dimensional Arrays	
5.3. Multidimensional Arrays	
5.4. Strings	
5.5. Arrays of Strings	
5.6. Pointers	
5.7. Pointers and Arrays	
6. Functions	08 Hrs
6.1. The general form of a function	
6.2. Creating functions	
6.3. Using Arguments (pass by value and reference)	
6.4. Using return	
6.5. Using functions in expressions	
6.6. Local Scope	
6.7. Global Scope	
6.8. Passing points and arrays to functions	
6.9. Passing command line arguments to main()	
6.10. Function Prototypes	
6.11. Recursion	
7. Classes and Objects	12 Hrs
7.1. The General Form of a Class	
7.2. Defining a Class and Creating Objects	
7.3. Adding Member Functions to a Class	
7.4. Constructors and Destructors	
7.5. Parameterized Constructors	
7.6. Arrays of Objects	

- 7.7. Initializing Object Arrays
- 7.8. Pointers to Objects
- 7.9. Inheritance Fundamentals
- 8. **The C++ I/O System** **06 Hrs**
 - 8.1. Understand I/O streams
 - 8.2. Know the I/O class hierarchy
 - 8.3. Format I/O by using ISO member functions
 - 8.4. Format I/O by using manipulators
 - 8.5. Open and close files
 - 8.6. Read and write text files
 - 8.7. Read and write binary files
 - 8.8. Use random access files I/O
- 9. **Software Design and Modularity** **08 Hrs**
 - 9.1. Elements of a good software design
 - 9.2. Decomposing larger problems into smaller problems
 - 9.3. Importance of modularity

RECOMMENDED BOOKS

- C++: A Beginner's Guide, Second Edition, Herbert Schildt
- A complete guide to programming C++ by Ulla Kirch-Prinz
- Learn to program with C++ by John Smiley

INSTRUCTIONAL OBJECTIVES

1. Introduction to Programming

- 1.1. Giving instructions to humans vs machines - Explain the difference in the type of instructions that can be given to a human vs a machine e.g. Washing Machines (timers, water level, etc.), Microwave (timers, heat level, etc.). Humans can process vague instructions whereas machines need exact, precise, and step-by-step instructions for performing a job. Example: Calculator is a complete computer. It has processor (the arithmetic operations) to do its processing, it has one number memory (M+ and MRC), input device (keyboard) and output device (LCD screen).
- 1.2. Programming Computers - Explain how computers are general-purpose “computing” devices that can perform any type of “calculations”, and send and receive signals to other devices (e.g. sensors, doors, other computers, lights, fans, cameras, etc.). Computers require exact instructions specifying all the steps to execute them.
- 1.3. Programming Languages - Explain that we need programming languages for giving instructions to the computers. They have exact grammar (syntax) like English.
- 1.4. Categories of Programming Languages – (Low and High Level) Explain why high-level languages were developed. Explain that large programs are complex and difficult to understand. We need high-level languages to make them easy to understand. Example: Assembly and C are low-level languages, show their sample programs. Java and Python are high-level languages, show their sample programs.

2. Introduction to C++ Fundamentals

- 2.1. Explain the structure of a C++ program
- 2.2. Introduce arithmetic operators
- 2.3. Describe what happens when input from the keyboard is read
- 2.4. Introduce if else statement using blocks in curly braces { }
- 2.5. Introduce while control statements with curly braces { }
- 2.6. Introduce float, int

3. Data Types and Operators

- 3.1. Introduce C++ data types, especially char, int, float, double, bool, and void. Explain the difference between signed, unsigned, long, and short. Tell them the range of values each type can hold.
- 3.2. Introduce numerical and string literals.
- 3.3. Explain in detail about variables and their initializations.
- 3.4. Introduce Arithmetic Operators and explain the precedence among them
- 3.5. Introduce Relational and Logical Operators
- 3.6. Explain in detail The Assignment Operator
- 3.7. Explain Implicit Type Conversion in Assignments

- 3.8. Explain Type Conversion in Expressions
- 3.9. Explain Casts, explicit.
- 3.10. Explain how Parentheses affect precedence in expressions

4. Program Control Statements

- 4.1. Explain the use of The if Statement
- 4.2. Explain the use of The if-else Statement
- 4.3. Explain the use of The if-else-if Statement
- 4.4. Explain the use of The switch Statement
- 4.5. Explain the use of The for Loop
- 4.6. Explain the use of The while Loop
- 4.7. Explain the use of The do while Loop

5. Arrays and Strings

- 5.1. Explain the use of One-Dimensional Arrays
- 5.2. Explain the use of Two-Dimensional Arrays
- 5.3. Explain the use of Multidimensional Arrays
- 5.4. Explain the use of Strings
- 5.5. Explain the use of Arrays of Strings

6. Functions

- 6.1. Explain the general form of a function
- 6.2. Explain Creating functions
- 6.3. Explain Using Arguments (pass by value and reference)
- 6.4. Explain Using return
- 6.5. Explain Using functions in expressions
- 6.6. Explain Local Scope
- 6.7. Explain Global Scope
- 6.8. Explain Passing points and arrays to functions
- 6.9. Explain Passing command line arguments to main()
- 6.10. Explain Function Prototypes
- 6.11. Explain Recursion

7. Classes and Objects

- 7.1. Explain The General Form of a Class
- 7.2. Explain Defining a Class and Creating Objects
- 7.3. Explain Adding Member Functions to a Class
- 7.4. Explain Constructors and Destructors
- 7.5. Explain Parameterized Constructors
- 7.6. Explain Arrays of Objects
- 7.7. Explain Initializing Object Arrays

- 7.8. Explain Pointers to Objects
- 7.9. Explain Inheritance Fundamentals

8. **The C++ I/O System**

- 8.1. Explain I/O streams
- 8.2. Explain the I/O class hierarchy
- 8.3. Explain Formatting I/O by using ios member functions
- 8.4. Explain Formatting I/O by using manipulators
- 8.5. Explain Opening and closing of files
- 8.6. Explain Reading and writing of text files
- 8.7. Explain Reading and writing binary files
- 8.8. Explain use of random access files I/O

9. **Software Design and Modularity**

- 9.1. Explain in detail the elements of a good software design. Simplicity, understandability, readability
- 9.2. Decomposing larger problems into smaller problems
- 9.3. Importance of modularity

LIST OF PRACTICALS

1. Practical 1

- Introduce general look and feel of Visual studio Community Edition 2022, especially build menu and Solution Explorer. Also practice Create Project of type “C++ Console App” using “create new solution”.
- Repeat creating a new solution and project. And walk through following steps.
 - a. Delete the contents of cpp file. Type the following code into the cpp file of the project created. Press “Build”>Build Solution to confirm if you typed it correctly.

```
#include <iostream>
using namespace std;
int main(){
    cout << "Salaam Pakistan from the Programmer of CIT!\n";
    return 0;
}
```

- b. Run the program using “Debug”>”Start Debugging”. Notice the output.
- c. What sense can you make out of each line. Write down after each line of the program using single line comment style // to tell what do you think this does.

2. Practical 2

- Do the following
 - a. Type the program.

<pre>#include <iostream> using namespace std; int main(){ int a; int b; int sum; a = 10;</pre>	<pre> b = 13 sum = a + b; cout << "The sum of "<<a<<" and "<<b<<" is: \n"<<sum<<"\n"; return 0; }</pre>
--	--

- b. Build the program. Remove any errors if you get.
- c. Run the program. Observe its output. Understand what each line is doing.
- d. Write description of each line, as you understand it, as comments using // on the right side of each line.
- e. Confirm if you wrote the comments correctly by building the program. Remove if you receive any error.
- f. Change values of a and b, and run the modified program to confirm if it still works.

3. Practical 3

- Do the following programming exercise.
 - Type and build the following program. Remove errors if any.

```
#include <iostream>
using namespace std;
int main(){
    int a;
    int b;
    int sum;
    float average;
    a = 10;
    b = 13;
    sum = a + b;

    average = a;
    average = average + b;
    average = average / 2; //this is divide
    operator
    cout << "The sum is: \n"<<sum<<"\n";
    cout << "The average is:\n";
    cout << average <<"\n";
    return 0;
}
```

--

- Build the program. Remove any errors if you get.
- Run the program. Observe its output. Understand what each line is doing.
- Write description of each line, as you understand it, as comments using // on the right side of each line.
- Confirm if you wrote the comments correctly by building the program. Remove if you receive any error.
- Change values of a and b, such that the output of the program becomes the following:

The sum is: 58 The average is:29

- Change values of a and b, such that the output of the program becomes the following:

The sum is: -8 The average is: -4

4. Practical 4

- Do the following programming exercise.
 - Type and build the following program that calculates the result of Tipu. Remove errors if any.

```
#include <iostream>
using namespace std;
int main(){
    int physics_marks = 70;
    int maths_marks = 55;
    int sum = physics_marks + maths_marks;
    float average = physics_marks;
    average = average + maths_marks;
    average = average / 2;

    cout <<
    "*****\n";
    cout << "Total marks are "<<sum<<"\n";
    cout << "The average marks are "<< average
    <<"\n";
    cout <<
    "*****\n";
    return 0;
}
```

- Without running the program, calculate what will be the output of this program.
- Run the program and confirm if your calculation is correct.
- Tipu has scored 80 marks in Urdu. Modify the program to include marks of Urdu.

- e. Tipu studies at Central Model School Samanabad. Modify the program such that a beautiful result card is printed on the screen. Hint: Add Tipu's full name and school name and city. You can also add borders.

5. Practical 5

- Do the following problems
 - a. Type the following program. Compile and remove errors. Run the program and write down what it is doing using comments.

```

-----
#include <iostream>
using namespace std;
int main(){
    int islamiat_marks = 70;
    int pakstudies_marks = 55;
    int computer_marks = 70;
    int maths_marks = 55;
    int physics_marks = 70;
    int chemisty_marks = 55;
    int Total = 100 * 6;
    int marks = chemisty_marks+ physics_marks
+ maths_marks;
    marks = marks + computer_marks +
pakstudies_marks;
    marks = marks + islamiat_marks;
    float percentage = (float) marks / Total;
    cout <<
"*****\n";
    cout << "Tipu has scored
"<<percentage<<"% marks with divison = ";
    if ( percentage > 80 ) {
        cout << "1st";
    }
    if ( percentage <= 80 && percentage > 65 ) {
        cout << "2nd";
    }
    if ( percentage <= 65 && percentage > 33 ) {
        cout << "3rd";
    }
    if ( percentage <= 33 ) {
        cout << "FAIL";
    }
    return 0;
}

```

- b. Change marks such that the division is changed.
- c. Put a breakpoint in line # 4. Start debugging. Use step over option to run the program step by step. Notice that the program does not display anything until you run the first cout line in stepped over.
- d. Using break point run the program again. Add watch to all variables including Total and islamiat_marks etc. Notice how each value changes at some point.

6. Practical 6

- Do the following
 - a. In Gilgit Baltistan province, the board has approved a new scheme using grades. Change the following program so that the following scheme is followed to display the grade of student.

Serial	Percentage Range	Grade
1	≥90	A+
2	≥80, <90	A
3	≥72, <80	A-
4	≥65, <72	B+

5	$\geq 58, < 65$	B
6	$\geq 51, < 58$	B-
7	$\geq 43, < 51$	C+
8	$\geq 38, < 43$	C
9	$\geq 30, < 38$	D
10	< 30	FAIL

```

#include <iostream>
using namespace std;
int main(){
    int islamiat_marks = 70;
    int pakstudies_marks = 55;
    int computer_marks = 70;
    int maths_marks = 55;
    int physics_marks = 70;
    int chemisty_marks = 55;
    int Total = 100 * 6;
    int marks = chemisty_marks+ physics_marks
+ maths_marks;
    marks = marks + computer_marks +
pakstudies_marks;
    marks = marks + islamiat_marks;
    float percentage = (float) marks / Total;
    cout << "*****\n";
    cout << "Tipu has scored
"<<percentage<<"% marks with divison = ";
    if ( percentage > 80 ) {
        cout << "1st";
    }
    if ( percentage <= 80 && percentage > 65 ) {
        cout << "2nd";
    }
    if ( percentage <= 65 && percentage > 33 ) {
        cout << "3rd";
    }
    if ( percentage <= 33 ) {
        cout << "FAIL";
    }
    return 0;
}

```

- Using breakpoints run the program again. Add watch to all variables including Total and islamiat_marks etc. Notice how each value changes at some point. Tell whether the program runs each line as you expected.
- Make 10 copies of the program file above. Modify marks of subjects in each copy, such that each of the given grades in the table is shown by exactly one program. Use step by step debugging to correct your logic if needed.

7. Practical 7

- Do the following
 - a. Type and build the following program.

```
#include <iostream>
using namespace std;
int main() {
    int num1, num2;
    // Prompt user for input of A
    cout << "Enter integer A: ";
    // Input a number from user and assign it to
    num1 variable
    cin >> num1;
    // Prompt user for input of B
    cout << "Enter integer B: ";
    // Input a number from user and assign it to
    num2 variable

    cin >> num2;
    // Compare and print result
    if (num1 > num2) {
        cout << "A is greater than B" << endl;
    }
    if (num1 < num2) {
        cout << "B is greater than A" << endl;
    }
    if (num1 == num2) {
        cout << "A and B are equal" << endl;
    }
    return 0;
}
```

- b. Read the program line by line and try to understand what this program is doing. Write description.
- c. Run the program and input numbers 457 and 90 to the program. Write down the output.
- d. Provide different values for A and B and see how the output of the program changes.
- e. Finally write down exactly three sets of input values, where each set produces a different output. Run the program to check the output.

8. Practical 8

- Do the following
 - a. Change to the program that calculates student grades you created in one of your previous labs. Change the logic as follows, the program should now take input from the user for marks of different subjects and then display grade in a nice format.
 - b. Run the modified grading program several times with different values of marks and observe that the grade is calculated correctly.
 - c. Run the program 10 times with different input values of marks such that each of the given grades in the grade table is output exactly once. Write down those value sets for each grade letter.

9. Practical 9

- Do the following
 - a. Type and build the following program

```
#include <iostream>
using namespace std;
int main() {
    int num, i;
    // Prompt user for input
    cout << "Enter an integer: ";
    cin >> num;
    // Print table header
    cout << "Multiplication Table of " << num <<
    endl;
    cout << "-----" << endl;

    // Loop from 1 to 20 using while loop
    i = 1;
    while (i <= 20) {
        cout << num << " x " << i << " = " << num *
        i << endl;
        i++; // Increment counter inside the loop
    }
    return 0;
}
```

- b. Run the program with different inputs, understand it, and note down the results. Especially note the effect of while. Use debugging breakpoints and watches to confirm your understanding.
- c. This program always prints the table up till 20. Modify the program such that it asks for the table length e.g. 20, and then print the table till that number.
- d. Bonus: Print a nicely formatted and beautiful table perfectly aligned from top to bottom

10. Practical 10

- Do the following
 - a. Type and build the following program

```

#include <iostream>
using namespace std;
int main() {
    float tipuHeight, sultanHeight;
    // Prompt user for input
    cout << "Enter Tipu's height: ";
    cin >> tipuHeight;
    cout << "Enter Sultan's height: ";
    cin >> sultanHeight;
    // Compare heights
    if (tipuHeight > sultanHeight) {
        cout << "Tipu is taller than Sultan." << endl;
    } else if (tipuHeight < sultanHeight) {
        cout << "Sultan is taller than Tipu." << endl;
    } else {
        cout << "Tipu and Sultan have the same height." << endl;
    }
    return 0;
}

```

- b. Notice use of else. Notice how it reduces the code and still achieves the correct result.
- c. Convert grading program such that it also uses else with every if. Confirm if the output is still correct for different values.
- d. Revise the program you coded in previous part as follows, check if the user inputs each subject's marks correctly. Each mark should be between 0 and 100, i.e. it cannot be negative or greater than 100. If the input is incorrect, display "wrong input" message and exit the program. Confirm if the working of your program is correct.

11. Practical 11

- Do the following
 - a. Write a program that takes an integer as input from the user and prints squares of all integers from 1 to the input. For example if the user enters 5, the output will print squares of all integers from 1 to 5 i.e. "1, 4, 9, 16, 25".
 - b. Write a program that takes an integer as input and prints all the integers from 0 to the input, for example if the input is 5 the output should be "0, 1, 2, 3, 4, 5".
 - c. Write a program that takes an integer as input and prints all the integers from the input to 0, for example if the input is 5 the output should be "5, 4, 3, 2, 1, 0".
 - d. Write a program that takes an integer as input and prints from 0 to the number and then back to 0. For example if the input is 5 the output should be "0, 1, 2, 3, 4, 5, 4, 3, 2, 1, 0".
 - e. Write a program that takes an integer as input and prints all the integers from 100 to the input. For example, if the user enters 45, the program prints integers from 100 to 45. If the user enters 138, the program prints integers from 100 to 138. Test your program by providing different inputs that are less than and greater than 100. What happens to the program when you provide input 100?

12. Practical 12

- Finding limits of integer type variables
 - Write a program to input an integer and then output the number. Run it to confirm it works. Now run it again but input 13762938754890274 number. See if it is working correctly. If not, write your guess why not.
 - The following program finds out the maximum value an integer variable can hold.

```
-----
#include <iostream>
using namespace std;

int main() {
    int num = 0;
    int prevNum = 0; // Stores the previous value
    of num

    // Loop to keep adding 1 until overflow
    while (true) {
        prevNum = num; // Save the previous value
        num++;         // Add 1 to num

        // Check for overflow using a comparison
        with the previous value
        if (num < prevNum) {
            cout << "Overflow occurred at: " <<
            prevNum << endl;
            break; // Exit the loop on overflow
        }
    }

    return 0;
}
-----
```

- Modify the program to find out the minimum value an integer can hold. Start with zero and keep subtracting 1 from it. You also have to adjust the if condition within the loop.
- Write a program to discover the maximum and minimum limits of unsigned int, long, unsigned long, char, unsigned char in order.

13. Practical 13

- Conversion of data types *Type and build the following program

```
-----
#include <iostream>
#include <limits> // for numeric_limits
using namespace std;
int main() {
    // Variable declarations with initial values
    float floatingNum = 3.14159;
    float floatingNum2 = 3.14159;
    int wholeNum = -321;
    int wholeNum2 = -321;
    long longNum = 1234567890123;
    long longNum2 = 1234567890123;
    unsigned int uintNum = 4294967295;
    unsigned int uintNum2 = 4294967295;
    cout << "Initial values:" << endl;
    cout << " float: " << floatingNum << endl;
    cout << " int: " << wholeNum << endl;
    cout << " long: " << longNum << endl;
    cout << " unsigned int: " << uintNum << endl <<
    endl;
    wholeNum2 = floatingNum; // float to int
(truncates)
    longNum2 = wholeNum; // int to long (no change)
    uintNum2 = wholeNum; // int to unsigned int (may
lose info)
    cout << " float to int: " << wholeNum2 << endl;
    cout << " int to long: " << longNum2 << endl;
    cout << " int to unsigned int: " << uintNum2 <<
    endl << endl;

    float floatNum2 = longNum; // long to float (may
lose precision)
    int intNum2 = uintNum; // unsigned int to int (may
lose info)

    cout << " long to float: " << floatNum2 << endl;
    cout << " unsigned int to int: " << intNum2 <<
    endl << endl;

    // Largest possible values (system-dependent)
    cout << "\nMaximum values for integer types:" <<
    endl;
    cout << " Max int: " <<
    numeric_limits<int>::max() << endl;
    cout << " Max long: " <<
    numeric_limits<long>::max() << endl;
    cout << " Max unsigned int: " <<
    numeric_limits<unsigned int>::max() << endl;
}
-----
```

```
numeric_limits<unsigned int>::max() << endl;    }
return 0;
```

- Assign different values to the variables and see the effect of assigning them to variables of other types. Notice how capacity of a variable effects the value that can be assigned to it.
- How many bytes int and long variables takes in memory? From the number of bytes calculate the largest value an int and long can hold. Note that unsigned int has the same size as int but the largest value it can hold is more than int. Calculate from the number of bytes of unsigned int what will be the largest value it can hold.
- Assign values to the variables in the program such that most of the conversions give wrong output values.

14. Practical 14

- Do the following
 - Write a program that converts feet into meters. Remember meters could be in fractions.
 - Write a program that prints a table of feet to meters (for 1 to N feet), where N is taken as input from the user.
 - Chaddar Gump, goes everywhere on foot. His 1 step = 1.6 feet, if he takes exactly 1100 steps to reach home from college, how many kilometers is his home from college? Write a program for Gump to convert steps in kilometers so that whenever he goes anywhere, he can find out how many kilometers he has traveled.

15. Practical 15

- Write build and run the programs for solving the following problem
 - The water tank in your house is rectangular in feet 6,6,5 (length, width, height). The water tank in your Chachu's house is cylindrical in feet 3,10 (radius, height). The flow rate of the water motor is used to fill the tank is 250 gallons per minute
 - Note the following
 - Volume of Rectangular Tank: $\text{Volume} = \text{length} * \text{width} * \text{height}$
 - Volume of Cylindrical Tank: $\text{Volume} = \pi * \text{radius}^2 * \text{height}$
 - 1 cubic foot = 7.48052 gallons
 - Time to Fill Tank: $\text{Time} = \text{Volume (in gallons)} / \text{Flow rate (in gallons/minute)}$
 - Your program should calculate the following
 - Volume of each water tank in cubic feet
 - Volume of each tank in gallons
 - Time to fill water tank in your house and in your chachu's house.
 - Which water tank takes less time to fill?
 - Match your results with other students
 - Can the motors fill the water tanks in 6 minutes?

16. Practical 16

- Write build and run the programs for solving the following problem
 - You are building a house on a 5 Marla plot. The plot size is 25 x 45 feet. The foundation has already been laid.

- b. Now you have to construct a boundary wall around the plot. Before you can do that you need to calculate the number of bricks required for constructing the boundary wall.
- c. Following information is available to you
 - i. Brick size is 9x4x3 inches (length, width, height)
- d. Calculate the number of bricks required for constructing the boundary wall which is 10 feet high.
- e. Now modify the program to take as input the following and calculate the bricks requirement
 - i. Plot dimensions
 - ii. Height of the boundary wall
 - iii. Brick dimensions

17. Practical 17

- Write build and run the programs for solving the following problem
 - a. Modify the bricks calculation program to calculate the following.
 - b. The brick suppliers supply in batches of 1000 bricks. How many batches will you order?
 - c. The price of one batch is PKR 1700. Calculate the cost for constructing the boundary wall.
 - d. For every 1000 bricks, you need 8 bags of cement. The price of one bag of cement is PKR 1300. Calculate the total cost of constructing the boundary wall.

18. Practical 18

- Write build and run the programs for solving the following problem
 - a. Following two programs are doing the exact same thing. Understand these programs and then map each statement of first program to a statement of the second program.

Program 1

```

-----
#include <iostream>
using namespace std;
int main() {
    int number, sum = 0;
    cout << "Enter a positive integer: ";
    cin >> number;
    int count = 1;
    while (count <= number) {
        sum += count;
        count++;
    }
    cout << "Sum of first " << number << "
integers is: ";
    cout << sum << endl;
    return 0;
}
-----
-----

```

Program 2

```

-----
#include <iostream>
using namespace std;
int main() {
    int number, sum = 0;
    cout << "Enter a positive integer: ";
    cin >> number;
    for (int count = 1; count <= number;
count++) {
        sum += count;
    }
    cout << "Sum of first " << number << "
integers is: ";
    cout << sum << endl;
    return 0;
}
-----

```

- b. Number Guessing Game: Develop a game where the user has to guess a randomly generated number. Use int to store the number and implement logic to

- check if the user's guess (also stored in int) is higher, lower, or correct. Provide hints to the user to improve the answer. Solve this problem using the while loop.
- Modify the Number Guessing Game program to use for loop instead of while loop. Compare that the modified project behaves exactly the same way as the previous one. Which loop structure do you like the most?

19. Practical 19

- Write build and run the programs for solving the following problem
 - Write a program that prints the following pattern on the screen with “*” characters


```
*****
*****
*****
****
***
**
*
```
 - Modify the program to allow for variable number of rows to be printed. Take input N from the users and print N rows.
 - Write a program that prints the following pattern on the screen with “*” characters


```
*
**
***
****
*****
*****
*****
```
 - Modify the program to allow for variable number of rows to be printed. Take input N from the users and print N rows.

20. Practical 20

- Write build and run the programs for solving the following problem
 - Understand the following program. Explain. Run and confirm if your understood correctly.

```
#include <iostream>
using namespace std;

#include <iostream>
#include <cmath>
#include <chrono>
#include <thread>
#include <windows.h>
#define M_PI 3.14
using namespace std;
// Constants
const double GRAVITY = 10.0; // m/s^2
const int TERMINAL_WIDTH = 80; // Adjust for
your terminal size
long long delay = 100;
void printGola(int x, int y) {
    y = 40 - y;
    COORD coord;
    coord.X = x;
```

```
    coord.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUT
TPUT_HANDLE), coord);
    cout << "o"; // Represent the cannonball with 'o'

    this_thread::sleep_for(chrono::milliseconds(delay));
    // Pause for a short period
}

int main() {
    delay = 100;
    int x, y;
    system("CLS");
    for (int x = 0; x < 40; x++) {
        y = x;
        printGola(x, y);
    }
    return 0;
}
```

--

- b. Modify above zig zag to touch top middle and then bottom right.
- c. Given speed and angle of a cannon gun (top) that fires the cannon (gola), calculate how high will it go, how far it will go, and modify the prev to animate firing.
 - i. The equation for horizontal distance is:
 - ii. $d = v^2 \sin(2\theta) / g$, where v is the velocity, θ is the angle of fire and g is acceleration due to gravity 10 m/s^2 .
 - iii. The equation for vertical distance is:
 - iv. $h = (v \sin(\theta))^2 / 2g$

21. Practical 21

- Write build and run the programs for solving the following problem
 - a. Modify the cannon gun game to include the following
 - i. Introduce Targets
 - ii. Make a simple scoring game
 - iii. Ends if maximum tries (10) or the target is hit.
 - b. Add menu to the game for starting, pausing, resuming and exiting the game

22. Practical 22

- Write build and run the programs for solving the following problem
 - a. <<Before snake game provide sample program that shows different output for arrow keys. Print“Left” “Right” etc. All the elements of the game should be in the sample program.>>
 - b. Create a snake game with the fixed-sized snake. The snake should move one step at a time with one arrow key.

23. Practical 23

- Write build and run the programs for solving the following problem
 - a. Modify the snake game you have developed to make the movement of the snake automatic without requiring arrow key input.
 - b. The snake dies if it touches boundary.

24. Practical 24

- Write build and run the programs for solving the following problem
 - a. Modify the snake game to add menu to it.
 - b. The menu should have the following options Start, Quit, Play Again

25. Practical 25

- Write build and run the programs for solving the following problem
 - a. Modify the snake game such that the snake grows by eating eggs. The eggs are placed randomly on the screen.
 - b. Further modify the game to set speed of snake as variable. Add a menu option for getting speed from the user.
 - c. Growth with eating eggs.
 - d. Bonus: Add coloring and sound to the game

26. Practical 26

- Write build and run the programs for solving the following problem
 - a. Read and understand the following program. Run the program again and again to see it's behavior

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
int main() {
    // Seed the random number generator with the
    current time
    srand(time(0));

    // Generate a random number between 0 and 9
    int randomNumber = rand() % 10;
    cout << "Random number: " <<
    randomNumber << endl;

    return 0;
}
```

- b. Build a simple Luddo game using the random number generator in place of throwing a dice.
 - i. There is only one path that runs around the edges of the screen.
 - ii. Player 1 Goti starts at the middle left of the screen. The previous square is Home of Player 1.
 - iii. Player 2 Goti starts at the middle right of the screen. The previous square is Home of Player 2.
 - iv. A player wins if it kills the other or if gets to it's home.
 - v. The game is played using a random number generated between 1 and 6 each time a player wants to move Goti.

27. Practical 27

- Write build and run the programs for solving the following problem
 - a. Understand the following program and explain what it is trying to achieve

```
#include <iostream>
using namespace std;
void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                // swap arr[j] and arr[j+1]
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

void printArray(int arr[], int size) {
    for (int i = 0; i < size; ++i)
        cout << arr[i] << " ";
    cout << endl;
}

int main() {
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr) / sizeof(arr[0]);
    cout << "Unsorted array: \n";
    printArray(arr, n);

    bubbleSort(arr, n);
    cout << "Sorted array: \n";
    printArray(arr, n);
    return 0;
}
```

-
- b. Modify the program for sorting in descending order
 - c. Modify the program that prints the student names and weights as follows:
 - i. Sorts students on weights in ascending order.

- ii. Sort students on names in ascending order.

28. Practical 28

- Write build and run the programs for solving the following problem
 - a. Run the following program:

```
-----
--
#include <iostream>
#include <string>
using namespace std;
int main() {
    string paragraph1 = "pakiStaN is A LAnD of BreAtHtaKIng bEAuty. FrOM the TOWeRIng PeAKs
Of tHe hiMAIAYAs tO thE SEREnE BEaCHES Of The Arablan sea, ThE cOUNtRy offErS a dIvERSE
Range OF lAnDscApeS. THE LUsh grEeN valLeYs, CAsCADIng wAtErfaLIS, AND crYstal-cLEAR
lAKES ArE a SiGHt TO BEhOID. tHe VibRaNT CULtUrE, WaRM hoSPitALiTy, aND RiCh HiSTOrY
ADD to tHe ChArM Of thIs BeAUtifuL NATION.";
    string paragraph2 = "pAkisTAn Is also hOMe to A rICH wILDLife. thE dIvERSE ECoSYSteMs
suPport A wIDe VARiety of FlorA AnD FAUna. fROM thE SNOW LEopaRDS in the nOrTheRN
moUntAiNs To tHe dolphInS in The CoaStaL WAtERS, The cOUNtrY OFFERs iNcrediBLE
oppOrtUNITiES foR WiLDLife ENtHUSIAstS. the vAsT DESERTS, fertIlE pLAINS, aNd mangROvE
fORESTS pRoVide HaBItaTs for CoUnTLEsS SpECiES.";

    cout << paragraph1 << endl << endl;
    cout << paragraph2 << endl;

    return 0;
}
-----
```

- b. Modify the program to correct the case of the text.
- c. Run the same program on the following text to see if it fixes the case correctly

```
-----
pAKiSTAN iS bLESSEd with a nETWoRk oF riveRS THAt nOuRish ThE LAND AND sUppOrt Its
PeOpLE. The inDus RiVer, ThE LIfeLINE oF thE cOUNTRy, fLOWs ThroUGh vasT PLaiNs,
cReaTiNg fErtile aGRiCULTUral LANd. OThER MaJoR rIvers lIkE The cHENAb, JheLUM, And ravi
cONtributE To ThE CouNtRy's wAtEr reSoURCES AnD HyDroelECtRic POWer gENERAtIon.
-----
```

29. Practical 29

- Write build and run the programs for solving the following problem
 - a. Given the following program

```
-----
#include <iostream>
using namespace std;
#include <iostream>
#include <cmath>
#include <chrono>
#include <thread>
#include <windows.h>
#define M_PI 3.14
using namespace std;
-----
```

```

// Constants
const double GRAVITY = 10.0; // m/s^2
const int TERMINAL_WIDTH = 80; // Adjust for your terminal size
long long delay = 100;
void printGola(int x, int y) {
    y = 40 - y;
    COORD coord;
    coord.X = x;
    coord.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
    cout << "o"; // Represent the cannonball with 'o'
    this_thread::sleep_for(chrono::milliseconds(delay)); // Pause for a short period
}
int main() {
    delay = 100;
    int x, y;
    system("CLS");
    for (int x = 0; x < 40; x++) {
        y = x;
        printGola(x, y);
    }
    return 0;
}
-----
--

```

- b. Modify the program and create AnimatGola() function. This function should change color of the Gola while in motion. Ensure that the program runs correctly.

30. Practical 30

Write build and run the programs for solving the following problem

- a. Modify the Luddo game and create the following functions
 - i. int RollDice() // This function rolls the dice and return a number
 - ii. AdvanceGoti(int n) // This function advances the Goti by n steps
- b. Ensure that Luddo game works correctly.

31. Practical 31

- Write build and run the programs for solving the following problem
 - a. Understand the program and write down what it is doing.

```
#include <iostream>
#include <string>
using namespace std;
string encrypt(string text) {
    string encrypted_text = "";
    for (char c : text) {
        if (isalpha(c)) {
            encrypted_text += static_cast<char>(c +
1);
        } else {
            encrypted_text += c;
        }
    }
    return encrypted_text;
}

string decrypt(string text) {
    string decrypted_text = "";
    for (char c : text) {
        if (isalpha(c)) {
            decrypted_text += static_cast<char>(c -
1);
        } else {
            decrypted_text += c;
        }
    }
    return decrypted_text;
}

int main() {
    string text = "This is a simple text to
encrypt";
    string encrypted_text = encrypt(text);
    string decrypted_text =
decrypt(encrypted_text);
    cout << "Original text: " << text << endl;
    cout << "Encrypted text: " << encrypted_text
<< endl;
    cout << "Decrypted text: " << decrypted_text
<< endl;
    return 0;
}
```

- b. Modify the program to use encoding by advancing ascii code by 5 (5 steps).
- c. Modify the program that gets input steps from the user and advance by that number of steps for encryption. Your program should correctly decrypt the text.
- d. Notice that if you use 13 step encryption, encrypting the text twice will decrypt it!

32. Practical 32

- Write build and run the programs for solving the following problem
 - a. Understand the following program. Explain. Run and confirm if you understood correctly. (Simple encryption with one character using XOR)

```
#include <iostream>
#include <string>
using namespace std;
string xor_encrypt(string text, char key) {
    string cipher = "";
    for (char c : text) {
        cipher += c ^ key;
    }
    return cipher;
}

string xor_decrypt(string cipher, char key) {
    string text = "";
    for (char c : cipher) {
```

```

        text += c ^ key;
    }
    return text;
}
int main() {
    string plaintext = "Hello, world!";
    char key = 'X';
    string ciphertext = xor_encrypt(plaintext, key);
    cout << "Ciphertext: " << ciphertext << endl;
    string decrypted_text = xor_decrypt(ciphertext, key);
    cout << "Decrypted text: " << decrypted_text << endl;
    return 0;
}

```

- b. Modify the program to make it menu based with the following options
- i. Press 1 to Provide key character
 - ii. Press 2 to Provide text to encrypt
 - iii. Press 3 to print text
 - iv. Press 4 to encrypt the text
 - v. Press 5 to print encrypted text
 - vi. Press 0 to exit

33. Practical 33

- Write build and run the programs for solving the following problem
 - c. Understand the following program. Explain. Run and confirm if you understood correctly. (Simple encryption with two character using XOR)

```

--
#include <iostream>
#include <string>
using namespace std;
string xor_encrypt(string text, string key) {
    string cipher = "";
    int key_len = key.length();
    for (int i = 0; i < text.length(); ++i) {
        cipher += text[i] ^ key[i % key_len];
    }
    return cipher;
}
string xor_decrypt(string cipher, string key) {
    string text = "";
    int key_len = key.length();
    for (int i = 0; i < cipher.length(); ++i) {
        text += cipher[i] ^ key[i % key_len];
    }
    return text;
}
int main() {
    string plaintext = "Hello, world!";
    string key = "XY";
    string ciphertext = xor_encrypt(plaintext,
    key);
    cout << "Ciphertext: " << ciphertext << endl;
    string decrypted_text =
    xor_decrypt(ciphertext, key);
    cout << "Decrypted text: " << decrypted_text
    << endl;
    return 0;
}
--

```

- d. Modify the program to make it menu based with the following options
- i. Press 1 to Provide key string
 - ii. Press 2 to Provide text to encrypt

- iii. Press 3 to print text
- iv. Press 4 to encrypt the text
- v. Press 5 to print encrypted text
- vi. Press 0 to exit

34. Practical 34

- Write build and run the programs for solving the following problem
 - a. Understand the following program. Explain. Run and confirm if you understood correctly.

```

#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main() {
    string text = "This is a simple text to be
written to a file.";
    ofstream outputFile("output.txt");
    if (outputFile.is_open()) {
        outputFile << text;
        outputFile.close();
        cout << "Text written to output.txt
successfully." << endl;
    } else {
        cout << "Error opening output file." <<
endl;
    }
    ifstream inputFile("output.txt");
    string line;
    if (inputFile.is_open()) {
        while (getline(inputFile, line)) {
            cout << line << endl;
        }
        inputFile.close();
    } else {
        cout << "Error opening input file." <<
endl;
    }
    return 0;
}

```

- b. Modify the string key encryption/decryption program to include following menu:
 - i. Press 1 to Provide key string
 - ii. Press 2 to Provide input file name to encrypt
 - iii. Press 3 to print input file contents
 - iv. Press 4 to encrypt the input file contents
 - v. Press 5 to print cipher (encrypted file contents)
 - vi. Press 6 to Provide output file name and write cipher to output file
 - vii. Press 0 to exit
 - c. Using the above program create secret files and share with your friends. Only you and your friend can decrypt the file and see what is inside!

35. Practical 35

- Write build and run the programs for solving the following problem
 - a. Understand the following program. Explain. Run and confirm if you understood correctly. (Chaining encryption with one character using XOR)

```

#include <iostream>
#include <string>
using namespace std;
string xor_encrypt(string text, char key) {
    string cipher = "";
    for (char c : text) {
        cipher += c ^ key;
    }
    return cipher;
}

```

```

}

string xor_decrypt(string cipher, char key) {
    string text = "";
    for (char c : cipher) {
        text += c ^ key;
    }
    return text;
}

int main() {
    string plaintext = "Hello, world!";
    char key = 'X';
    string ciphertext = xor_encrypt(plaintext, key);
    cout << "Ciphertext: " << ciphertext << endl;
    string decrypted_text = xor_decrypt(ciphertext, key);
    cout << "Decrypted text: " << decrypted_text << endl;
    return 0;
}

```

--

- b. Modify the program to read the input file and write the cipher to the output file using menu options.

36. Practical 36

- Write build and run the programs for solving the following problem
 - a. Modify the chaining encryption program to use a string as a key instead of a single character.

37. Practical 37

- Write build and run the programs for solving the following problem
 - a. Create a simulator of a Naan Shop Queue. Customers are being added to the queue with the following details:
 - i. Name
 - ii. Order (e.g. 1 Sadi Roti, 2 Khamiri, 1 Roghni Naan, 3 Aloo Walay Naan etc.)
 - b. Customer can be added to the queue using menu option by providing Name and Order information.
 - c. The Complete Queue should be printed and visible on the screen at all times
 - d. It's a first come first served arrangement. A customer at the front of the queue will be served. Once the order is complete, the program should print Name and Order details for the customer and remove the customer from the queue.
 - e. The time required to complete an order is calculated from the time required for individual items.
 - i. Sada Roti - First 1 minute, rest 10 seconds each
 - ii. Khamiri Roti - First 1.5 minutes, rest 15 seconds each
 - iii. Sada Naan - First 1.5 minutes, rest 15 seconds each
 - iv. Roghani Naan - First 2 minutes, rest 20 seconds each
 - v. Aaloo Wala Naan - First 3 minutes, rest 20 seconds each
 - vi. Chicken Naan - First 3 minutes, rest 20 seconds each
 - vii. Keemay Wala Naan - First 3 minutes, rest 25 seconds each
 - f. It should also be displayed on the screen which customer is being served and which item is being backed.

38. Practical 38

- Write build and run the programs for solving the following problem
 - a. Create the Zero Kata (Tic Tac Toe) using object oriented programming.
 - b. Given below is the skeleton code

```
--
#include <iostream>
using namespace std;
const int SIZE = 3;
char board[SIZE][SIZE] = {{',',',','},
                           {'',',',','},
                           {'',',',','}};
char currentPlayer = 'X';
void displayBoard() {
    // Implement code to display the board
}
bool isBoardFull() {
    // Implement code to check if the board is
    full
    return false;
}
bool checkWin(char player) {
    // Implement code to check if the player has
    won
    return false;
}
void makeMove(int row, int col) {
    // Implement code to make a move
}

int main() {
    bool gameOver = false;
    while (!gameOver) {
        displayBoard();
        // Get player input for row and column
        makeMove(row, col);
        if (checkWin(currentPlayer)) {
            cout << "Player " << currentPlayer << "
            wins!" << endl;
            gameOver = true;
        } else if (isBoardFull()) {
            cout << "It's a draw!" << endl;
            gameOver = true;
        }
        currentPlayer = (currentPlayer == 'X') ?
        'O' : 'X';
    }
    return 0;
}
```

- a. Explanation:
 - i. The board array represents the Tic-Tac-Toe board.
 - ii. currentPlayer keeps track of the current player ('X' or 'O').
 - iii. displayBoard function will print the current state of the board.
 - iv. isBoardFull checks if the board is completely filled.
 - v. checkWin checks if the current player has won.
 - vi. makeMove updates the board with the player's move.
- b. Students can:
 - i. Implement the functions displayBoard, isBoardFull, checkWin, and makeMove.
 - ii. Add input validation for player moves.
 - iii. Enhance the game with features like player names, turn-based play, and AI opponents.

39. Practical 39

- Write build and run the programs for solving the following problem
 - a. Create the game of chess using object oriented programming.
 - b. Given below is the skeleton code

```
#include <iostream>
#include <vector>
using namespace std;
enum PieceType { PAWN, ROOK, KNIGHT,
BISHOP, QUEEN, KING };
enum PieceColor { WHITE, BLACK };
class Piece {
public:
    PieceType type;
    PieceColor color;
    int row, col;
    bool hasMoved;
    Piece(PieceType type, PieceColor color, int row,
int col) {
        // Initialize piece attributes
    }
    virtual bool isLegalMove(int newRow, int
newCol) = 0;
    virtual void move(int newRow, int newCol) = 0;
};
class Pawn : public Piece {
public:
    Pawn(PieceColor color, int row, int col) :
Piece(PAWN, color, row, col) {}
    bool isLegalMove(int newRow, int newCol)
override {
        // Implement pawn move logic
        return false;
    }
    void move(int newRow, int newCol) override {
        // Implement pawn move action
    }
};
// Other piece classes (Rook, Knight, Bishop,
Queen, King)
class Board {
public:
    vector<vector<Piece*>> board;
    Board() {
        // Initialize board with pieces
    }
    bool isOccupied(int row, int col) {
        // Check if position is occupied
        return false;
    }
    Piece* getPieceAt(int row, int col) {
        // Return piece at given position
        return nullptr;
    }
    bool movePiece(int startRow, int startCol, int
endRow, int endCol) {
        // Move piece from start to end position
        return false;
    }
};
class Player {
public:
    PieceColor color;
    vector<Piece*> pieces;
    Player(PieceColor color) {
        // Initialize player with pieces
    }
    void makeMove() {
        // Implement player move logic
    }
};
class Game {
public:
    Board board;
    Player player1, player2;
    PieceColor currentTurn;
    Game() {
        // Initialize game
    }
    void startGame() {
        // Start the game
    }
    void playTurn() {
        // Play a turn
    }
    bool isGameOver() {
        // Check if game is over
        return false;
    }
};
int main() {
    // Create a game instance and start playing
    return 0;
}
```

c. Explanation:

- i. This code provides a basic skeleton for a chess game using OOP.
- ii. The Piece class is an abstract base class for all chess pieces.
- iii. Specific piece classes inherit from Piece and implement their own move logic.
- iv. The Board class represents the chessboard and handles piece movement.
- v. The Player class represents a player and manages their pieces.
- vi. The Game class orchestrates the gameplay.
- d. Students can:
 - i. Implement the missing methods in the Piece, Board, Player, and Game classes.
 - ii. Add more attributes and methods as needed.
 - iii. Create additional classes for features like move history, etc.
 - iv. Design the user interface.

40. Practical 40

- Do the following
 - a. Understand the following program and write down what it is doing:

```
#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
void writeTextFile() {
    ofstream outputFile("textFile.txt");
    if (outputFile.is_open()) {
        outputFile << "This is a text file.\n";
        outputFile << "We can write formatted data:\n";
        outputFile << fixed << setprecision(2);
        outputFile << "Pi: " << 3.14159 << endl;
        outputFile.close();
        cout << "Text file written successfully.\n";
    } else {
        cout << "Error opening output file.\n";
    }
}
void readTextFile() {
    ifstream inputFile("textFile.txt");
    if (inputFile.is_open()) {
        string line;
        while (getline(inputFile, line)) {
            cout << line << endl;
        }
        inputFile.close();
        cout << "Text file read successfully.\n";
    } else {
        cout << "Error opening input file.\n";
    }
}
void writeBinaryFile() {
    ofstream outputFile("binaryFile.bin", ios::binary);
```

```

if (outputFile.is_open()) {
    int number = 42;
    double pi = 3.14159;
    char character = 'A';
    outputFile.write(reinterpret_cast<char*>(&number), sizeof(number));
    outputFile.write(reinterpret_cast<char*>(&pi), sizeof(pi));
    outputFile.write(&character, sizeof(character));
    outputFile.close();
    cout << "Binary file written successfully.\n";
} else {
    cout << "Error opening output file.\n";
}
}

void readBinaryFile() {
    ifstream inputFile("binaryFile.bin", ios::binary);
    if (inputFile.is_open()) {
        int number;
        double pi;
        char character;
        inputFile.read(reinterpret_cast<char*>(&number), sizeof(number));
        inputFile.read(reinterpret_cast<char*>(&pi), sizeof(pi));
        inputFile.read(&character, sizeof(character));
        cout << "Number: " << number << endl;
        cout << "Pi: " << pi << endl;
        cout << "Character: " << character << endl;
        inputFile.close();
        cout << "Binary file read successfully.\n";
    } else {
        cout << "Error opening input file.\n";
    }
}

void randomAccessFile() {
    fstream file("randomAccessFile.dat", ios::in | ios::out | ios::binary);
    if (file.is_open()) {
        int data[3] = {10, 20, 30};
        file.write(reinterpret_cast<char*>(data), sizeof(data));
        file.seekp(sizeof(int)); // Move to the second integer
        int newData = 40;
        file.write(reinterpret_cast<char*>(&newData), sizeof(newData));
        file.seekg(0); // Move to the beginning
        int readData[3];
        file.read(reinterpret_cast<char*>(readData), sizeof(readData));
        cout << "Data after modification: ";
        for (int i = 0; i < 3; ++i) {
            cout << readData[i] << " ";
        }
        cout << endl;
        file.close();
        cout << "Random access file operations successful.\n";
    } else {
        cout << "Error opening file.\n";
    }
}

```

```
    }  
}  
int main() {  
    writeTextFile();  
    readTextFile();  
    writeBinaryFile();  
    readBinaryFile();  
    randomAccessFile();  
    return 0;  
}
```

--

- b. Now create a program that input student full names and their city and write in the text file.
- c. Create a program
 - i. That input student full names, their city, age, and height and write in the text file with proper format and spacing.
 - ii. Read the same file and remove the formatting and store in arrays. Remove the extra spaces etc. while reading the file.
 - iii. Display all the information read from the file

CIT-141 Computer & Peripheral Maintenance

Total Contact Hours

Theory	0	T	P	C
Practical	96	0	3	1

A major course dealing with the main component of the microcomputer motherboard, it's internally Attached components and the peripherals. It is designed to provide student with skills and Knowledge to maintain, diagnose and repair computer main system board and peripherals. It covers Concepts and methods of preventive maintenance, fault diagnose of motherboard, power supply, Memory, storage device, PCB, printers, monitors and other externally attached component of Computer system. This has laboratory component.

Learning Outcomes: The students will be able to

- Understand the concepts and methods of computer system maintenance
- Apply skills and knowledge to maintain PC main system component and peripherals.
- Understand the operational principles PC power supplies.
- Assemble and disassemble printers, scanners, monitor, mouse, keyboard and cards.
- Diagnose and repair faulty PC main system board and peripherals.
- Demonstrate patience and motivation in repairing minute computer components.

List of Practical

1. Introduction & Safety Procedures: Overview of course and safety procedures for handling computer hardware.
2. Identify Computer Components: Open a computer case and identify major components (motherboard, hard disk ,CPU, RAM, power supply, etc.).
3. Assembling a Computer: Assemble a computer from individual components.
4. Disassembling a Computer: Safely disassemble a computer and reassemble it.
5. Connecting Peripherals: Connect and configure peripheral devices (printers, scanners, external drives).
6. Troubleshooting Peripherals: Diagnose and resolve common issues with peripheral devices.
7. Introduction to OS Installation: Overview of different operating systems and installation processes.
8. Install Windows OS: Perform a clean installation of Windows operating system.
9. Install Linux OS: Perform a clean installation of a popular Linux distribution (e.g., Ubuntu).
10. Dual-Boot Setup: Configure a dual-boot system with Windows and Linux.
11. BIOS/UEFI Settings: Access and configure BIOS/UEFI settings.

12. OS Configuration: Configure basic operating system settings and preferences.
13. Install Essential Software: Install essential software and utilities (e.g., antivirus, productivity tools).
14. Configuration of Multimedia
15. Installation and Setup of ink jet Printer
16. Identification and Disassembly/Assembly of ink jet Printer
17. Identification and Disassembly/Assembly of LCD/ LED and its Interface Card
18. Troubleshooting, Preventive and Corrective Maintenance of LCD/ LED

RECOMMENDED BOOKS

- CompTIA A+ 220-801 and 220-802 Authorized Cert Guide, Third Edition, Mark Edward Soper, David L. Prowse and Scott Mueller.
- Microprocessors And Interfacing 2E, DOUGLA V Hall by McGraw-Hill Education
- The Complete A+ Guide to PC Repair Fifth Edition Update (5th Edition), Cheryl Schmidt
- A Guidebook in PC Hardware, Maintenance And Repair Ent 284, William A. Lloyd by PRINCE GEORGE'S COMMUNITY COLLEGE
- Computer Hardware Technicians CH111, 5 CD-ROM set by NCSA Academy USA Computer Hardware Technicians

اسلامیات

تدریس مقاصد:

عمومی مقاصد: طالب علم یہ جان سکے کہ آیات قرآنی کی روشنی میں مومن کے اوصاف کیا ہیں

() قرآن مجید () منتخب آیات قرآنی

خصوصی مقاصد:

() قرآنی آیات کا ترجمہ

() قرآنی آیات کی تشریح

() قرآنی آیات کی روشنی میں ایک مومن کے اوصاف بیان کر سکے

() قرآنی آیات میں بیان کردہ مومن کے اوصاف اپنے اندر پیدا کر سکے

احادیث نبوی:

عمومی مقاصد:

() احادیث کی روشنی میں اسلامی اخلاقی اقدار (انفرادی و اجتماعی) سے آگاہ ہو سکے

خصوصی مقاصد:

() احادیث کا ترجمہ بیان کر سکے

() احادیث کی تشریح کر سکے

() احادیث کی روشنی میں اسلام کی اخلاقی اقدار کی وضاحت کر سکے

() احادیث کی دی گئی تعلیمات کے مطابق اپنی زندگی گزار سکے

سیرت طیبہ:

() عمومی مقاصد: حضور ﷺ کی سیرت طیبہ کے بارے میں جان سکے

خصوصی مقاصد:

() حضور ﷺ کی ابتدائی زندگی اختصار کے ساتھ بیان کر سکے

() حضور ﷺ کی ہجرت کا واقعہ بیان کر سکے

() حضور ﷺ کی مدنی زندگی اختصار سے بیان کر سکے

() حضور ﷺ کی بطور معلم خصوصیات بیان کر سکے

() حضور ﷺ کی بطور سربراہ بیان کر سکے

حضور ﷺ بحیثیت

() خطبہ حجۃ الوداع۔ انسانی حقوق کا اعلامیہ حقوق انسانی پالیسی 2018 کے بارے جان سکے

اسلامی معاشرہ:

عمومی مقاصد: اسلامی معاشرہ کی خصوصیات سے آگاہی حاصل کر سکے

خصوصی مقاصد:

() اسلامی معاشرہ کا معنی و مفہوم کر سکے

() اسلامی معاشرہ کی امتیازی خصوصیات بیان کر سکے

() اسلامی معاشرہ میں عدل و احسان کی اہمیت بیان کر سکے

() تبلیغ کے لغوی معنی بیان کر سکے

() تبلیغ کی اہمیت اور ضرورت بیان کر سکے

() جہاد کے لفظی و اصطلاحی معنی بیان کر سکے

() جہاد کی اہمیت بیان کر سکے

() جہاد اور قتل میں فرق بیان کر سکے

() جہاد کی مختلف اقسام بیان کر سکے

() لفظ مسجد کی تعریف بیان کر سکے

() مسجد کی سابقہ حیثیت کو بحال کرنے کے بارے میں اقدامات کو مان سکے

() انسانی حقوق کی معاشی اور معاشرتی ترقی میں اہمیت اور کردار بیان کر سکے

اسلامی ریاست:

عمومی مقاصد:

() اسلامی ریاست کی خصوصیات بیان کر سکے

خصوصی مقاصد:

() ریاست کی تعریف بیان کر سکے

() اسلامی ریاست میں طرز حکومت سے آگاہی حاصل کر سکے

() اسلامی ریاست کی خصوصیات بیان کر سکے

حصہ دوم
مطالعہ پاکستان

تدریس مقاصد:

تحریک پاکستان:

عمومی مقاصد:

() قیام پاکستان کے اسباب و تحریک کو بیان کر سکے

خصوصی مقاصد:

() قومیت کے مفہوم کو بیان کر سکے

() دو قومی نظریہ کی تعریف و توضیح کر سکے

() دو قومی نظریہ کی اہمیت بیان کر سکے

() ہندوستانی مسلمانوں کی محرومیوں کو بیان کر سکے

() قومی تشخص کو بحال رکھنے کے لیے مسلمانان ہند کی مساعی بیان کر سکے

() آزادی ہند اور قیام پاکستان علامہ اقبال اور قائد اعظم کی مسایم کریں

() قیام پاکستان سے مستقبل اسلامی مملکت کے قیام کے لیے مسلم عوام کی کوششوں کو بیان کر سکے

() مسلم لیگ کے قیام پاکستان کے لیے جدوجہد بیان کر سکے

(غیر مسلم طلباء کے لیے)

ٹی پی سی
1 0 1
کل وقت 20 گھنٹے

نصاب اخلاقیات

سال دوم

موضوعات:

معاشرتی اقدار بلحاظ ہمسایہ۔ قوم۔ قومی سطح۔ شہری سطح۔ صنعتی اداروں کی سطح۔ ضروریات۔ ورثہ

() حقوق و فرائض

() قوت برداشت

() قوت ارادی

() لگن و جذبہ

() وسیع النظری

() بے غرضی

() انسانی دوستی

() حفاظتی شعور

() پاس آزادی

() کامل آگاہی

() تغیرات کو قبول کرنا

() خود شناسی

() انسانی حقوق کا اعلامیہ حقوق انسانی پالیسی 2018

() انسانی حقوق کا معاشرتی اور معاشی ترقی میں کردار

() حقوق آگاہی / معلومات تک رسائی۔ ملازم پیشہ خواتین کا تحفظ

نصاب اخلاقیات

سال دوم

تدریس مقاصد:

طالب علم اخلاقیات کی اہمیت و ضرورت سے آگاہ ہو سکے اور بیان کر سکے

خصوصی مقاصد: طالب علم اس قابل ہو کہ

- () موضوعات کا مطلب بیان کر سکے
- () عملی زندگی سے مثالوں کی نشاندہی کر سکے
- () اپنی شخصیت اور معاشرے پر موضوعات کے مطابق اثرات پیدا کرنے کے طریقے بیان کر سکے
- () اعلیٰ اخلاقی اقدار میں سے قوت برداشت - قوت ارادی - لگن و جذبہ - وسیع النظری - بے غرض - انسانی دوستی شعور - پاس آزادی - کامل آگاہی اور خود شناسی کی اہمیت
- () اخلاقیات سے متصف ہو کر قومی خدمت بہتر طور پر انجام دے سکے
- () انسانی حقوق کا اعلامیہ حقوق انسانی کی پالیسی 2018 کے بارے میں جان سکے
- () انسانی حقوق کا معاشرتی اور معاشی ترقی میں اہمیت اور کردار بیان کر سکے
- () حقوق آگاہی / معلومات تک رسائی کے قانون کو جان سکے
- () ملازم پیشہ خواتین کا تحفظ کو اخلاقیات کا حصہ تصور کرے

TTQ 211/Civic-211 TARJAMA TUL QURAN/
Civics

T	P	C
1	0	1
Total Contact Hours:		
Theory:		32
Practical:		0

As per syllabus of BISE from PCTB Book

MATH-233 APPLIED MATHEMATICS-II

Total Contact Hours

Theory	96	T	P	C
Practical	0	3	0	3

Pre-requisite: Must have completed Mathematics-I.

AIMS At the end of the course, the students will be able to:

Solve problems of Calculus, Laplace Transformation and Fourier Series, and develop mathematical skills and logical perceptions in the use of mathematical instruments.

COURSE CONTENTS

- 1. FUNCTIONS & LIMITS. 6 hours**
 - 1.1 Constant & Variable Quantities
 - 1.2 Functions & their classification
 - 1.3 The concept of Limit
 - 1.4 Limit of a Function
 - 1.5 Fundamental Theorems on Limit
 - 1.6 Some important Limits
 - 1.7 Problems
- 2. DIFFERENTIATION 6 hours**
 - 2.1 Increments
 - 2.2 Differential Coefficient or Derivative
 - 2.3 Differentiation ab-initio or by first Principle
 - 2.4 Geometrical Interpretation of Differential Coefficient
 - 2.5 Differential Coefficient of X^n and $(ax + b)^n$
 - 2.6 Three important rules
 - 2.7 Problems
- 3. DIFFERENTIATION OF ALGEBRAIC FUNCTIONS 9 hours**
 - 3.1 Explicit Functions
 - 3.2 Implicit Functions
 - 3.3 Parametric forms
 - 3.4 Problems
- 4. DIFFERENTIATION OF TRIGONOMETRIC FUNCTIONS 6 hours**
 - 4.1 Differential Coefficient of $\sin x$, $\cos x$, $\tan x$ from first principle.
 - 4.2 Differential Coefficient of $\operatorname{cosec} x$, $\sec x$, $\cot x$
 - 4.3 Differential Coefficient of Inverse trigonometric functions.
 - 4.4 Problems.

5.	DIFFERENTIATION OF LOGARITHMIC & EXPONENTIAL FUNCTIONS	6 hours
5.1	Differentiation of $\ln x$	
5.2	Differentiation of $\log a^x$	
5.3	Differentiation of a^x	
5.4	Differentiation of e^x	
5.5	Problems	
6.	RATE OF CHANGE OF VARIABLES	6 hours
6.1	Increasing and decreasing functions	
6.2	Maxima and Minima	
6.3	Criteria for maximum & minimum values	
6.4	Methods of finding maximum & minimum	
6.5	Rate measure	
6.6	Slope of a line	
6.7	Velocity and acceleration	
6.8	Problems	
7.	INTEGRATION(SIMPLE BASIC RULES)	9 hours
7.1	Concept	
7.2	Fundamental Formulas	
7.3	Important Rules	
7.4	Problems	
8.	METHODS OF INTEGRATION	9 hours
8.1	Integration by substitution	
8.2	Integration by parts	
8.3	Problems	
9.	DEFINITE INTEGRALS	6 hours
9.1	Properties	
9.2	Application to area	
9.3	Problems	
10.	DIFFERENTIAL EQUATIONS	6 hours
10.1	Introduction	
10.2	Order and Degree	
10.3	First order Differential Equation of 1st degree.	
10.4	Solution of problems	
10.5	Problems	
11.	LAPLACE TRANSFORMATIONS	9 hours
11.1	Laplace Transformations	
11.2	Inverse Laplace Transformations	
11.3	Problems.	
12.	FOURIER SERIES.	9 hours

- 12.1 Introduction
- 12.2 Periodic Functions
- 12.3 Even and Odd Functions
- 12.4 Problems

13. STATISTICS

9 hours

- 13.1 Concept of mean, median and mode
- 13.2 Standard Deviation
- 13.3 Laws of probability
- 13.4 Problems

RECOMMENDED BOOKS:-

1. Text Book of Math-233, developed by Curriculum Section, Academics wing TEVTA and published by National Book Foundation (NBF).

MGM-211 Business Communication

Total Contact Hours

Theory	32	T	P	C
Practical	0	1	0	1

Second course on English language focusing on business communication. It aims to develop communication skills as applied in business and commerce such as the writing and business correspondence. It covers oral communication and art of listening, interviewing, and report writing among others. It is a pure class discussion.

Learning Outcomes: The students will be able to

- Understand the basic principles of good and effective business writing in commercial and industrial fields.
- Use the English language effectively for communication in business
- Apply knowledge and skill to write business communication with confidence and ease.
- Write legibly in handwriting and compose communication documents with correct formats.
- Appreciate the usefulness of written language

COURSE OUTLINE

Contents	Hours
1. Communication Process.	06 Hrs
1.1. Purposes of communication	
1.2. Communication process	
1.3. Distortions in communication	
1.4. Consolidation of communication	
1.5. Communication flow	
1.6. Communication for self development	
2. Oral Communication Skills	06 Hrs
2.1. Significance of Speaking	
2.2. Verbal and Non-verbal Messages	
2.3. Strategic Steps of Speaking	
2.4. Characteristics of Effective Oral Messages	
2.5. Communication Trafficking	
2.6. Oral Presentation	

3. Questioning Skills	03 Hrs
3.1. Nature and Types of Questions	
3.2. Characteristics of a Good Questions	
3.3. Questioning Strategy	
4. Listening Skills	05 Hrs
4.1. Principles of Active Listening	
4.2. Skills of Active Listening	
4.3. Barriers to Listening	
4.4. Reasons of Poor Listening.	
4.5. Giving Feedback.	
5. Interviewing Skills	03 Hrs
5.1. Significance of Interviews	
5.2. Characteristics of Interviews	
5.3. Activities in an Interviewing Situation	
5.4. Types of Interviews.	
5.5. Interviewing Strategy	
6. Report Writing	03 Hrs
6.1. Goals of Report Writing	
6.2. Report Format	
6.3. Types of Reports	
6.4. Report Writing Strategy	
7. Reading Comprehension	02 Hrs
7.1. Reading Problems	
7.2. Four Reading Skills	
8. Group Communication	04 Hrs
8.1. Purposes of Conducting Meetings	
8.2. Planning a Meeting	
8.3. Types of Meetings	
8.4. Selection of a Group for Meeting	
8.5. Group Leadership Skills	
8.6. Running a Successful Meeting	
8.7. Active Participation Techniques	

RECOMMENDED BOOKS

- Effective Business Communication and Report Writing, Sh. Ata-ur-Rehman.
- Technical Reporting, Ulman J.N. Could JR..

INSTRUCTIONAL OBJECTIVES

1. Communication Process

- 1.1. Understand the communication process
- 1.2. State the benefits of two way communication
- 1.3. Describe a model of communication process
- 1.4. Explain the major communication methods used in organization
- 1.5. Identify the barriers to communication and methods of overcoming these barriers
- 1.6. Identify misconceptions about communication

2. Oral Communication Skills

- 2.1. Understand the process of oral communication
- 2.2. Identify speaking situations with other peoples
- 2.3. Identify the strategy steps of speaking
- 2.4. Identify the characteristics of effective speaking
- 2.5. State the principles of one-way communication
- 2.6. State the principles of two-way communication
- 2.7. Identify the elements of oral presentation skills
- 2.8. Determine the impact of non-verbal communication on oral communication

3. Questioning Skills

- 3.1. Determine the uses of questioning skills and clarify information in the oral communication process
- 3.2. Identify different types of questions
- 3.3. Determine the purpose of each type of question and its application
- 3.4. Identify the hazards to be avoided when asking questions
- 3.5. Demonstrate questioning skills

4. Listening Skills

- 4.1. Demonstrate the use of active listening skill in the oral communication process
- 4.2. State the principles of active listening
- 4.3. Identify skills of active listening
- 4.4. Identify barriers to active listening
- 4.5. State the benefits of active listening
- 4.6. Demonstrate listening skills.
- 4.7. Explain the importance of giving and receiving feed back

5. Interview Skills

- 5.1. Determine the appropriate interview type for the specific work-related situation and conduct a work-related interview
- 5.2. State the significance of interviews
- 5.3. State the characteristics of interviews
- 5.4. Explain the activities in an interviewing situation
- 5.5. Describe the types of interviews

- 5.6. Explain the interviewing strategy
- 5.7. Prepare instrument for a structured interview

6. Report Writing

- 6.1. Prepare a report out-line, based on subject matter and audience
- 6.2. Identify the different types of reports
- 6.3. Determine when to use an informal or formal report presentation
- 6.4. Identify the stages of planning a report
- 6.5. Identify the parts of a report and choose the parts appropriate for each type of report
- 6.6. Draft a report outline

7. Reading Comprehension

- 7.1. Demonstrate reading comprehension
- 7.2. Identify major reading problems
- 7.3. Identify basic reading skills
- 7.4. State methods of previewing written material
- 7.5. Identify methods of concentration when reading
- 7.6. Demonstrate reading comprehension

8. Group Communication

- 8.1. Understand the principles of group communications
- 8.2. State the purpose and characteristics of major types of meeting
- 8.3. Explain responsibilities of a meeting/committee
- 8.4. Identify problems likely to be faced at meeting and means to overcome these problems
- 8.5. Distinguish between content and process at meetings
- 8.6. Explain the key characteristics of a good group facilitator

CIT- 203 Data Communication & Computer Networks

Total Contact Hours

Theory	64	T	P	C
Practical	96	2	3	3

Introductory course on data communication and computer networks. It aims to provide the students conceptual tools to understand the design and implementation of data communication as applied to computer networks. It discusses the layered model. It covers communication, media, WAN, LAN, Internetworking, protocols, network management and troubleshooting. This comes with practical component to complement classroom discussion.

Learning Outcomes: The students will be able to

- Understand basic concepts and principles of data communication as applied to computer networking & Internet Protocols
- Synthesize the different approaches of data communication in networking and assess their effectiveness in implementation
- Troubleshoot and diagnose network faults and correct them
- Network Services and Network media devices
- Assemble cabling system of a network
- Recognize the inherent problems in networking and appreciate the solutions of the problems

COURSE OUTLINE

Contents	Hours
1. Introduction to computer Networks	10 Hrs
1.1. Introduction to Network	
1.2. Types of Networks	
1.3. Introduction to transmedia	
1.4. Cables and Connectors	
1.5. Cable Types	
1.6. Connector Types	
1.7. Cable Management	
2. Connecting Devices	08 Hrs
2.1. Modems	
2.2. Hubs and Repeaters	
2.3. Bridges, Routers and Gateways	
2.4. Wireless Communication	
2.5. Wireless Access point	

2.6. Wireless Controller

3. Network Model 15 Hrs

- 3.1. Introduction to Network Model
- 3.2. OSI Model
- 3.3. Layered task in OSI
- 3.4. TCP/IP Model
- 3.5. Difference between TCP and UDP
- 3.6. Layered task in TCP/IP
- 3.7. IP Addressing

4. Network Troubleshooting 05 Hrs

- 4.1. Testing Media connectivity testing & troubleshooting the connectivity of Media and devices
- 4.2. Network Testing Tools

5. Network Services 08 Hrs

- 5.1. Introduction to Network Services
- 5.2. DHCP
- 5.3. DNS
- 5.4. NTP

6. Hierarchical Network Design 10 Hrs

- 6.1. LAN Architecture
- 6.2. Topologies
- 6.3. Ethernet, Fast Ethernet and gigabit Ethernet
- 6.4. WLAN Architecture
- 6.5. Layers of Networks
- 6.6. Access Layer
- 6.7. Distribution Layer
- 6.8. Core Layer

7. Introduction of Data Communication & Multiplexing 08 Hrs

- 7.1. Introduction
- 7.2. Data Transmission
- 7.3. Analog Transmission
- 7.4. Digital Transmission
- 7.5. Signal Impairment
- 7.6. Telephony and Wireless Communication
- 7.7. Multiplexing
- 7.8. TDM
- 7.9. FDM

RECOMMENDED BOOKS

- Networking Essentials, Joe Casad and Dan Newland, Techmedia
- Computer Networks, 2Ed, Andrew S. Tanenbaum, Prentice Hall
- Data and Computer Communications, 4Ed., William Stallings, MacMillan
- Local Area Networks, 2Ed, Peter Hodos
- An introduction to Local Area Networks, Greg Nunemacher
- Networking Explained, Gallo
- Networking Essentials, 2Ed, Microsoft Press

INSTRUCTIONAL OBJECTIVE

1. Introduction to computer Networks

- 1.1. Explain the introduction to Network
- 1.2. Explain the types of Networks
- 1.3. Under the Introduction to transmedia
- 1.4. Explain the detail of cables and Connectors
- 1.5. Explain Cable Types
- 1.6. Describe Connector Types
- 1.7. State the Cable Management

2. Connecting Devices

- 2.1. Explain detail on Modems
- 2.2. Explain detail on Hubs and Repeaters
- 2.3. Explain detail on Bridges, Routers and Gateways
- 2.4. Explain detail on Wireless Communication.
- 2.5. Describe Wireless Access point
- 2.6. Describe Wireless Controller

3. Network Model

- 3.1. Detail explain on Introduction to Network Model
- 3.2. Understand importance of OSI Model
- 3.3. Explain the Layered task in OSI
- 3.4. Explain the TCP/IP Model
- 3.5. Difference between TCP and UDP
- 3.6. Explain the Layered task in TCP/IP
- 3.7. Detail explain on IP Addressing

4. Network Troubleshooting

- 4.1. Detail explain on Testing Media connectivity testing & troubleshooting the connectivity of Media and devices
- 4.2. Detail explain on Network Testing Tools

5. Network Services

- 5.1. Explain detail Introduction to Network Services
- 5.2. Explain detail DHCP
- 5.3. Explain detail DNS
- 5.4. Explain detail NTP

6. Hierarchical Network Design

- 6.1. State the LAN Architecture
- 6.2. State the Topologies
- 6.3. Ethernet, Fast Ethernet and gigabit Ethernet

- 6.4. Explain the WLAN Architecture
- 6.5. Explain detail Layers of Networks
- 6.6. State the Access Layer
- 6.7. Explain detail Distribution Layer
- 6.8. Explain detail State the Core Layer

7. Introduction of Data Communication & Multiplexing

- 7.1. Introduction
- 7.2. State the Data Transmission
- 7.3. Explain the Analog Transmission
- 7.4. Explain the Digital Transmission
- 7.5. State the Signal Impairment
- 7.6. Explain detail Telephony and Wireless Communication
- 7.7. Explain detail Multiplexing
- 7.8. Explain detail TDM
- 7.9. Explain detail FDM

List of Practical's

1. Basic LAN & Wireless Setup
2. Configure Static and Dynamic IP addresses.
3. Identifying various hardware devices of a network
4. Identifying network interface card (Wired & Wireless)
5. Make a Straight/Crossover cable using UTP /STP
6. Patch Panel with network switches.
7. Make a Female I/O and attach with face plates
8. Router Configuration & Wireless Network Setup
9. Connecting Wireless Access Point/Wireless Extender.
10. Configuring Omada devices and Configuration
11. Configure Mac Filtering on omada and routers
12. Design of a Local Area NW of computers
13. Installing and configuring a NW printer
14. Troubleshooting NW connectivity
15. Splicing of fiber optics cable and also use its connector
16. LAN to fiber converters to extend network.
17. Network Troubleshooting with Logs

CIT- 273 Digital Logic Design

Total Contact Hours

Theory	64	T	P	C
Practical	96	2	3	3

In this course, students will gain a foundational understanding of digital circuit design, including key techniques and fundamental concepts. They will explore various forms of number representation in digital electronics and master the skills to convert between these formats. The curriculum covers the implementation and design of both combinational and sequential logic circuits, equipping students with the ability to perform simple logical operations and build complex circuits. Additionally, students will delve into sequential circuits and the storage mechanisms of different memory systems, gaining insights into their functioning and conversion systems.

Learning Outcomes: The students will be able to

- Learn basic techniques for the design of digital circuits and fundamental concepts used in the design of digital systems.
- Understand common forms of number representation in digital electronic circuits and to be able to convert between different representations.
- Implement simple logical operations using combinational logic circuits
- Design combinational logic circuits, sequential logic circuits.
- Impart to student the concepts of sequential circuits, enabling them to analyze
- learn the storage mechanism of various memories and conversion system.

COURSE OUTLINE

Contents	Hours
1. Number Systems, Number Representations, and Codes	05 Hrs
1.1. Key concepts and Overview	
1.2. Digital vs. Analog	
1.3. Digital Design Overview (from Transistor to Super Computer)	
1.4. Design Methodologies	
1.5. Number Systems (Decimal, Binary, Octal, Hexadecimal)	
1.6. Base Conversions	
1.7. Signed Binary Number	
1.8. Binary Arithmetic (Addition, Subtraction)	
1.9. Binary Codes	

2. Boolean algebra, Functions, and Minimization.

07 Hrs

- 2.1. Basic Definitions
- 2.2. Basic Properties and Theorems of Boolean algebra
- 2.3. Boolean Functions, Simplification of Boolean Expressions
- 2.4. Canonical and Standard Forms
- 2.5. Logic Operators, Digital Logic Gates, Universal Gates
- 2.6. Positive and Negative Logic
- 2.7. K-maps

3. Combinational Logic Circuits

10 Hrs

- 3.1. Introduction of combinational circuits
- 3.2. Design Procedure of combinational circuits
- 3.3. Adders
- 3.4. Subtractors
- 3.5. Code Conversion
- 3.6. Parity Generator and Checker
- 3.7. Four-Bit Binary Parallel Adder
- 3.8. Four-bit Binary Parallel Subtractor
- 3.9. Four-bit Binary Parallel Adder/Subtractor
- 3.10. Magnitude Comparator
- 3.11. Decoders and Encoders
- 3.12. Multiplexers or Data Selectors
- 3.13. DE multiplexers or Data Distributors

4. Sequential Logic Circuits

10 Hrs

- 4.1. Introduction
- 4.2. Flip-flops
- 4.3. Types of Flip-flops
- 4.4. Clocked S-R Flip-flop
- 4.5. Clocked D Flip-flop
- 4.6. J-K Flip-flop
- 4.7. T Flip-flop
- 4.8. Triggering of Flip-flops
- 4.9. Multivibrators.

5. Registers **08 Hrs**

- 5.1. Introduction
- 5.2. Shift Register
- 5.3. Serial-in–Serial-out Shift Register
- 5.4. Serial-in–Parallel-out Register
- 5.5. Parallel-in–Serial-out Register
- 5.6. Parallel-in–Parallel-out Register
- 5.7. Universal Register
- 5.8. Shift Register Counters
- 5.9. Counters

6. COUNTERS A/D AND D/A CONVERSION **09 Hrs**

- 6.1. Introduction
- 6.2. Digital-to-Analog Converters (DAC)
- 6.3. Specification of D/A Converters
- 6.4. An Example of a D/A Converter
- 6.5. Analog-to-Digital Converters
- 6.6. Specification of an A/D Converter
- 6.7. An Example of an A/D Converter IC

7. LOGIC FAMILY **09 Hrs**

- 7.1. Introduction
- 7.2. Characteristics of Digital IC
- 7.3. Bipolar Transistor Characteristics
- 7.4. Resistor-Transistor Logic (RTL)
- 7.5. Diode Transistor Logic (DTL)
- 7.6. Transistor Logic (TTL)
- 7.7. Emitter-Coupled Logic (ECL)
- 7.8. Integrated-Injection Logic (I²L)
- 7.9. Metal Oxide Semiconductor (MOS)

8. MEMORY **06 Hrs**

- 8.1. Discuss Memory Technologies.
- 8.2. Discuss General Memory Operation.
- 8.3. Describe Memory Considerations
- 8.4. List Types of Memories
- 8.5. Discuss applications in Digital systems.

RECOMMENDED BOOKS

1. Digital Principles al Principles and Logic Design By A. SAHA & N. MANNA
INFINITY SCIENCE PRESS LLC Hingham, Massachusetts, New Delhi.
2. Digital Design, Fourth Edition, By M. Morris Mana

INSTRUCTIONAL OBJECTIVES

1. INTRODUCTION TO DIGITAL SYSTEMS-

- 1.1 Key concepts and Overview
- 1.2 Digital vs. Analog
 - 1.2.1 Define digital and analog signals with examples
 - 1.2.2 How analog signal convert into digital signal
- 1.3 Digital Design Overview (from Transistor to Super Computer)
- 1.4 Design Methodologies
 - 1.4.1 Top-down Design Methodology
 - 1.4.2 Bottom-up Design Methodology
 - 1.4.3 Middle-out Design Methodology
- 1.5 Number Systems (Decimal, Binary, Octal, Hexadecimal)
- 1.6 Base Conversions
 - 1.6.1 Conversion from Binary to Decimal Number System
 - 1.6.2 Conversion from Decimal to Binary Number System
 - 1.6.3 Conversion from Octal to Binary Number System
 - 1.6.4 Conversion from Binary to Octal Number System
 - 1.6.5 Conversion from Binary to Hexadecimal Number System
 - 1.6.6 Conversion from Hexadecimal to Binary Number System
 - 1.6.7 Fractional Conversion
- 1.7 Binary Arithmetic
 - 1.7.1 Binary (Addition, Subtraction, Multiplication, Division)
 - 1.7.2 Subtraction operation (using the 1's, 2's, r's and the $(r - 1)$'s complement)
 - 1.7.3 BCD Addition, BCD Subtraction
- 1.8 Signed Binary Number
 - 1.8.1 Define signed binary number/integer
 - 1.8.2 Different ways to represent Signed number/integer
- 1.9 Binary Codes
 - 1.9.1 Weighted Binary Codes (BCD Code or 8421 Code, 84-2-1 Code, 2421 Code)
 - 1.9.2 No weighted Binary Codes (Excess-3 Code, Gray Code)
 - 1.9.3 Conversion of decimal numbers into weighted and non-weighted codes)
- 1.10 DC Electrical Circuit Fundamentals
 - 1.10.1 Electrical Resistance, Power
 - 1.10.2 Circuit simplification by combining resistors

2. Boolean Algebra, Functions, and Minimization...

- 2.1 Basic Definitions
 - 2.1.1 Define Closer, Associative Law, Commutative Law, Identity Element, Inverse, Distributive Law)

- 2.2 Basic Properties and Theorems of Boolean algebra
 - 2.2.1 Principle of Duality
 - 2.2.2 DE Morgan's Theorem and Other Important Theorems
- 2.3 Boolean Functions
 - 2.3.1 Define Boolean function
 - 2.3.2 How Boolean function express algebraically
 - 2.3.3 How Boolean function represented by Truth Table
- 2.4 Simplification of Boolean Expressions
 - 2.4.1 Perform Simplification of Boolean Expressions using laws, theorems and truth tables
- 2.5 Canonical and Standard Forms
 - 2.5.1 Minterm and Maxterm
 - 2.5.2 Deriving a Sum of Products (SOP) Expression from a Truth Table
 - 2.5.3 Deriving a Sum of Products (SOP) Expression from a Truth Table
- 2.6 Logic Operators
 - 2.6.1 Null, AND, Inhibition, Equivalence, Complement
 - 2.6.2 Exclusive-OR, OR, NOR, NAND
- 2.7 Digital Logic Gates
 - 2.7.1 AND, OR, Inverter or NOT, Buffer, NAND, NOR, X-OR, X-NOR)
- 2.8 Universal Gates
 - 2.8.1 Realization of Logic Functions by NAND Gates
 - 2.8.2 Realization of Logic Functions by NOR Gates
- 2.9 Positive and Negative Logic
- 2.10 Karnaugh Maps
 - 2.10.1 Two-Variable Karnaugh Maps
 - 2.10.2 Three-Variable Karnaugh Maps
 - 2.10.3 Four - Variable Karnaugh Maps
 - 2.10.4 Don't-Care Combinations

3. Combinational Logic Circuits

- 3.1 Introduction of combinational circuits
- 3.2 Design Procedure of combinational circuits
- 3.3 Adders
 - 3.3.1 Design of Half-adders
 - 3.3.2 Design of Full-adders
- 3.4 Subtractors
 - 3.4.1 Design of Half-subtractors
 - 3.4.2 Design of Full-subtractors
- 3.5 Code Conversion
 - 3.5.1 Design of Binary-to-gray Converter
 - 3.5.2 Design of BCD-to-excess-3 Code Converter

- 3.6 Parity Generator and Checker
 - 3.6.1 Describe Parity Generator and Checker
- 3.7 Four-Bit Binary Parallel Adder
 - 3.7.1 Explain and draw the Four-Bit Binary Parallel Adder
- 3.8 Four-bit Binary Parallel Subtractor
 - 3.8.1 Explain and draw the Four-bit Binary Parallel Subtractor
- 3.9 Four-bit Binary Parallel Adder/Subtractor
 - 3.9.1 Explain and draw the Four-bit Binary Parallel Adder/Subtractor
- 3.10 BCD Adder
 - 3.10.1 Explain and design BCD Adder
- 3.11 Magnitude Comparator
 - 3.11.1 Define and Explain Magnitude Comparator with example
- 3.12 Decoders and Encoders
 - 3.12.1 Define decoder's circuit and its applications
 - 3.12.2 Design 3-to-8 line decoder
 - 3.12.3 Define Encoder's circuit and its applications
 - 3.12.4 Explain 8-to-1 line encoder circuit
- 3.13 Multiplexers or Data Selectors
 - 3.13.1 Define multiplexer
 - 3.13.2 Describe data selectors
 - 3.13.3 Explain 4-to-1 line multiplexer
- 3.14 Demultiplexers or Data Distributors
 - 3.14.1 Define De-multiplexer
 - 3.14.2 Describe data distributors
 - 3.14.2.1 Explain 1-to-8 line De-multiplexer

4. Sequential Logic Circuits

- 4.1 Introduction
 - 4.1.1 Define Sequential Circuits
 - 4.1.2 Explain the block diagram of Sequential Circuits
 - 4.1.3 Describe the difference between Combinational and Sequential Circuits
- 4.2 Flip-flops
 - 4.2.1 Describe the difference between Flip Flop and Latch's
 - 4.2.2 Define and Explain Types of Flip-flops
 - 4.2.2.1 S-R (Set-Reset) Flip-flop (based on NOR and NAND Gates)
- 4.3 Clocked S-R Flip-flop
- 4.4 Clocked D Flip-flop
- 4.6 J-K Flip-flop
- 4.7 T Flip-flop
- 4.8 Triggering of Flip-flops
 - 4.8.1 Level Triggering of Flip-flops

- 4.8.2 Edge-triggering of Flip-flops
- 4.9 Multivibrators.
 - 4.9.1 555 Timer as Monostable Multivibrator (One Shot)
 - 4.9.2 555 Timer as Astable Multivibrator (Free Running)
- 5. Registers and Counters**
 - 5.1 Introduction
 - 5.1.1 Define Register and n-bit Registers
 - 5.2 Shift Register
 - 5.2.1 Describe the purpose of shift register and its use
 - 5.3 Define and Explain Serial-in–Serial-out Shift Register
 - 5.3.1 Shift-right Register
 - 5.3.2 Shift-left Register
 - 5.4 Define and Explain Serial-in–Parallel-out Register
 - 5.5 Define and Explain Parallel-in–Serial-out Register
 - 5.6 Define and Explain Parallel-in–Parallel-out Register
 - 5.7 Universal Register
 - 5.7.1 Define ‘Unidirectional Shift Register’.
 - 5.7.2 Define ‘Bidirectional Shift Register’.
 - 5.8 Shift Register Counters
 - 5.8.1 Define and explain Ring Counter
- 5.9 Counters
 - 5.9.1 Define counters and its Categories
 - 5.9.2 Define Asynchronous and Synchronous Counter
- 6. COUNTERS A/D AND D/A CONVERSION**
 - 6.1 Introduction
 - 6.2 Digital-to-Analog Converters (DAC)
 - 6.2.1 Explain Weighted-resistor D/A Converter
 - 6.2.2 Explain R-2R Ladder D/A Converter
 - 6.3 Specification of D/A Converters
 - 6.3.1 Define Resolution, Linearity
 - 6.3.2 Define Accuracy, Settling Time and Temperature Sensitivity
 - 6.4 Analog-to-Digital Converters
 - 6.4.1 Describe Parallel Comparator A/D Converter
 - 6.4.2 Describe Counting A/D Converter
 - 6.4.3 Describe A/D Converter Using Voltage-to-Frequency Conversion
 - 6.4.4 Describe A/D Converter using Voltage-to-Time Conversion
 - 6.6 Specification of an A/D Converter
 - 6.6.1 Define Range of input voltage
 - 6.6.2 Define Input impedance, Accuracy, Conversion time
 - 6.6.3 Define Format of digital output

7. LOGIC FAMILY

7.1 Introduction

7.2 Characteristics of Digital IC

7.2.1 Define the most important properties of various logic families

Propagation delay (speed of operation), Power dissipation, Fan in, Fan out.

Noise immunity, Operating temperature, Power supply requirement

Current and voltage parameters.

7.3 Define and Introduction Bipolar Transistor Characteristics

7.4 Define and Introduction Resistor-Transistor Logic (RTL)

7.5 Define and Introduction Diode Transistor Logic (DTL)

7.6 Define and Introduction Transistor Logic (TTL)

7.7 Define and Introduction Emitter-Coupled Logic (ECL)

7.8 Define and Introduction Integrated-Injection Logic (I²L)

7.9 Define and Introduction Metal Oxide Semiconductor (MOS)

8- MEMORY

8.1 Discuss Memory Technologies.

8.2 Discuss General Memory Operation.

8.3 Describe Memory Considerations

8.4 List Types of Memories

8.4.1 Describe ROM

8.4.2 Describe RAM

8.4.3 Describe SRAM

8.4.4 Describe DRAM

8.4.5 Describe PLDs

8.4.6 Describe Magnetic and Optical Memories

8.5 Discuss applications in Digital systems.

LIST OF PRACTICAL

1. Practices For The Conversions Of One Base To Another Base
2. Study Basic Logic Gates and Their Functions (7400 ,7402 ,7404 ,7408 ,7432 ,7486)
3. Simplification of Boolean Functions
4. Assemble Binary Adder and Subtractor (half and full) Circuit and Verify Operations.
5. Design and implement 4-bit Parallel Adder/ subtractor using IC 7483.
6. Design and Implementation of 4-bit Magnitude Comparator using IC 7485.
7. Design And Implementation of 4-bit Code Convertor
 - 7.1 Binary to gray code converter
 - 7.2 Gray to binary code converter
 - 7.3 BCD to excess-3 code converter
 - 7.4 Excess-3 to BCD code converter
8. Design and implement 16-bit odd/even parity checker generator-using IC 74180.
9. Design and Implementation of Multiplexer and Demultiplexer
10. Design and Implementation of Encoder and Decoder
11. To analyze the operation of BCD to 7-segment decoder
12. Realize the following flip-flops using NAND Gates.
 - 12.1 Clocked SR Flip-Flop
 - 12.2 JK Flip-Flop.
13. Assemble a Monostable Multivibrator with the help of 555 Timer.
14. Assemble a Astable Multivibrator with the help of 555 Timer.
15. Construction and Verification of 4 Bit Ripple Counter
16. Design and Implementation of 3 Bit Synchronous Up/Down Counter
17. Design and Implementation of Shift Register using IC7474

CIT- 222 Operating System

Total Contact Hours

Theory	32	T	P	C
Practical	96	1	3	2

This hands-on course offers a comprehensive introduction to Linux OS, covering system structure, bash scripting, and essential administrative tasks. Participants will develop practical skills in user and resource management, backup automation, and deployment strategies. The curriculum includes virtual environment and container management with Docker and Kubernetes, and remote VM management on AWS, including network rules and security. The course emphasizes software installation, terminal productivity, and secure management using PKI keys, culminating in a capstone project that integrates all acquired skills.

Learning Outcomes: The students will be able to

At the end of the course, the students are expected to be able to exhibit

- Proficient Linux System Administration Manage users, permissions, resources, and automate tasks with bash scripting.
- Advanced Container and Cloud Management: Deploy and manage virtual environments using Docker, Kubernetes, and AWS, including network security.
- Effective Terminal Productivity and Security: Install and configure development tools, optimize terminal usage, and utilize PKI keys for secure management.

COURSE OUTLINE

Contents	Hours
1. What is Operating System	05 Hrs
1.1. Processes of operating system	
1.2. Privilege access rights and users	
1.3. Memory	
1.4. Storage	
1.5. Network	
1.6. Vms	
1.7. Cloud	
2. Processes and Threads	06 Hrs
2.1. Concept	
2.2. Comparison	
2.3. Context switching	
2.4. Multiprogramming	
2.5. Multithreading	

2.6. IPC

3. Memory	06 Hrs
3.1. Paging architecture	
3.2. Allocation	
3.3. Virtual memory	
4. Disk and File system	05 Hrs
4.1. Disk structure	
4.2. Drive vs partition vs file system	
4.3. Inode allocation structure	
4.4. Linked with fat allocation	
4.5. Fragmentation	
5. Cloud	10 Hrs
5.1. VM	
5.2. Containerization	
5.3. Comparison with virtual environments	
5.4. Infrastructure as code	
5.5. Virtual network	
5.6. Virtual routers	
5.7. Pki	

References

1. "The Linux Command Line: A Complete Introduction" by William Shotts.
2. "Docker Deep Dive" by Nigel Poulton.
3. "Kubernetes: Up and Running: Dive into the Future of Infrastructure" by Kelsey Hightower, Brendan Burns, and Joe Beda.
4. AWS Documentation and Training: <https://aws.amazon.com/documentation/>.
5. Operating system concepts Abraham Silberschatz

INSTRUCTIONAL OBJECTIVE

1. What is an Operating System

- 1.1. Processes
- 1.2. Privilege
- 1.3. Access rights and users
- 1.4. Memory
- 1.5. Storage
- 1.6. Network
- 1.7. VMs
- 1.8. Cloud

2. Processes and Threads

- 2.1. Concept
- 2.2. Comparison
- 2.3. Context switching
- 2.4. Multiprogramming
- 2.5. Multithreading
- 2.6. IPC

3. Memory

- 3.1. Paging architecture
- 3.2. Allocation
- 3.3. Virtual memory

4. Disk and File system

- 4.1. Disk structure
- 4.2. Drive vs partition vs file system
- 4.3. Inode allocation structure
- 4.4. Linked with fat allocation
- 4.5. Fragmentation

5. Cloud

- 5.1. Vm
- 5.2. Containerization
- 5.3. Comparison With Virtual Environments
- 5.4. Infrastructure As Code
- 5.5. Virtual Network And Virtual Routers
- 5.6. Pki

LIST OF PRACTICALS

1. Linux OS Basics: Overview of Linux distributions, File system hierarchy, Basic commands (ls, cd, pwd, mkdir, rm, etc.)
2. User and Group Management: Adding, modifying, and deleting users, Understanding user and group permissions, Managing groups
3. File Permissions and Ownership: File permission basics (read, write, execute), Changing permissions (chmod), Changing ownership (chown)
4. Sudo and Root Access: Understanding sudo and root user, Configuring sudoers file
5. Introduction to Bash Scripting: Writing and executing basic bash scripts, Variables, conditionals, and loops
6. Advanced Bash Scripting: Functions, arrays, and error handling in bash scripts
7. Automating User Management: Scripting user creation and deletion, Managing user permissions via scripts
8. Scripting User Monitoring: Tracking user activity, Logging user actions
9. System Resource Monitoring: Monitoring CPU, memory, and disk usage, Using tools like top, htop, and iostat
10. Scripting Resource Monitoring: Automating resource monitoring scripts, Generating resource usage reports
11. Backup Strategies: Understanding backup types (full, incremental, differential), Tools for backups (rsync, tar, etc.)
12. Automating Backups with Scripts: Writing backup scripts, Scheduling backups with cron jobs
13. Deployment Basics: Introduction to deployment strategies, Basic deployment scripts
14. Virtual Environment Basics: Understanding virtualization concepts, Setting up VirtualBox and Vagrant
15. Docker Introduction: Understanding containers, Basic Docker commands and Dockerfile
16. Docker Compose and Networking: Multi-container applications with Docker Compose, Networking in Docker
17. AWS Introduction: Overview of AWS services, Setting up an AWS account
18. EC2 Instances and Network Management: Launching and managing EC2 instances, SSH access using PKI keys, Configuring network rules and security groups
19. Installing Development Tools: Installing and configuring Git, Jira, etc., Managing software dependencies
20. Review of typical commands

CIT-283 Database Management System

Total Contact Hours

Theory	64	T	P	C
Practical	96	2	3	3

Databases are part of our everyday life. Whether we are accessing our bank accounts, paying bills, searching the web, or calling a call center, our requests are usually handled by a database management system. This course aims to shift the focus from "computation" to "information" and covers key concepts of organizing, storing, retrieving, and querying data. It includes the basics of database architecture, management, and systems, as well as the principles of database design and techniques for developing database applications.

Learning Outcomes: The students will be able to

- Understanding Basic Concepts of Data and Databases
- Exploring Database Models and Architectures
- Mastering Relational Database Management Systems
- Utilizing SQL for Database Manipulation
- Implementing Normalization and Advanced Database Design

COURSE CONTENTS

Contents	Hours
1. Introduction to Data and Information.	06 Hrs
1.1. Traditional File-Based Systems	
1.2. File-Based Approach	
1.3. Limitations of the File-Based Approach	
1.4. Introducing the Database	
1.5. Structure of Database	
1.6. Uses of DBMS	
1.7. Advantages and Disadvantages of DBMS	
1.8. Components of DB Applications	
1.9. Introduction to DBMS	
1.10. Role in Database Environment	
1.11. DB Tools; Microsoft Access, MySQL, SQL Server, Oracle	
2. Database Environment	08 Hrs
2.1. Database Models	
2.1.1. Object-Based Data Models	

- 2.1.2. Record-Based Data Models
 - 2.1.3. Physical Data Models
 - 2.1.4. Conceptual Modelling
 - 2.1.5. NoSQL Graph Database
- 2.2. Database architecture
 - 2.2.1. Traditional Two-Tier Client–Server Architecture
 - 2.2.2. Three-Tier Client–Server Architecture
 - 2.2.3. N-Tier Architectures
- 2.3. Database Languages
 - 2.3.1. Data Definition Language
 - 2.3.2. Data Manipulation Language
 - 2.3.3. Data Control Language
- 2.4. Introduction to Cloud based databases
- 2.5. Introduction to distributed databases

3. Relational Database Management System

06 Hrs

- 3.1. Database Entities and Attributes
- 3.2. Database Relations & Their Types
- 3.3. Relational Keys
- 3.4. Database Schema
- 3.5. Integrity Constraints
- 3.6. Entity integrity & referential integrity
- 3.7. Relations of Primary and secondary key, Cascading
- 3.8. Referencing in Database

4. Relational Algebra and SQL

06 Hrs

- 4.1. Relational Algebra
- 4.2. Built-in Functions
- 4.3. Unary Operations and Binary operations
- 4.4. Set Operations
- 4.5. Joins Operations
- 4.6. Types of Joins
- 4.7. Division Operation
- 4.8. Aggregation and Grouping Operations

5. Structuring Database

08 Hrs

- 5.1. Data Definition Language
- 5.2. Data Manipulation Language
- 5.3. Insertion, Deletion and Updation
- 5.4. Truncation & Concatenation
- 5.5. Simple Queries
 - 5.5.1. Sorting Results (ORDER BY Clause)
 - 5.5.2. Using the SQL Aggregate Functions
 - 5.5.3. Grouping Results (GROUP BY Clause)
- 5.6. Subqueries
 - 5.6.1. SELECT clause
 - 5.6.2. FROM Clause
 - 5.6.3. WHERE Clause
 - 5.6.4. HAVING Clause
 - 5.6.5. ANY and ALL
 - 5.6.6. Multi-table Queries
 - 5.6.7. EXISTS and NOT EXISTS
 - 5.6.8. Combining Result Tables (UNION, INTERSECT, EXCEPT)

6. Introduction to Database

06 Hrs

- 6.1. Data Definition Language
- 6.2. Data Manipulation Language
- 6.3. Insertion, Deletion and Updating
- 6.4. Truncation & Concatenation
- 6.5. Simple Queries
- 6.6. Sorting Results (ORDER BY Clause)
- 6.7. Using the SQL Aggregate Functions
- 6.8. Grouping Results (GROUP BY Clause)
- 6.9. Subqueries
- 6.10. SELECT clause
- 6.11. FROM Clause
- 6.12. WHERE Clause
- 6.13. HAVING Clause
- 6.14. ANY and ALL
- 6.15. Multi-table Queries
- 6.16. EXISTS and NOT EXISTS
- 6.17. Combining Result Tables (UNION, INTERSECT, EXCEPT)

7. Normalization of Database Tables	06 Hrs
7.1. Need for Normalization	
7.2. First Normal Form	
7.3. Second Normal Form	
7.4. Third Normal Form	
7.5. Boyce-Codd Normal Form (BCNF)	
8. Database Life Cycle (DBLC)	06 Hrs
8.1. Database Initial Study	
8.2. Database Design	
8.3. Database Design Strategies	
8.4. Centralized versus Decentralized Design	
9. Entity Relationship (E-R) Modelling	08 Hrs
9.1. Basic Modelling Concepts	
9.2. Degrees of Data Abstraction	
9.3. Association and Cardinality	
9.4. Relationship Participation	
9.5. Composite Entities, Entity Super types and subtypes	
9.6. Enhanced Entity Relationship Diagram	
9.7. Transform ER/EER to Relational Model	
10. Views, Indexes and Transaction Management	04 Hrs
10.1. Introduction to Views & Their Types	
10.2. Introduction to Indexing	
10.3. Introduction of Triggers	
10.4. What is a Transaction?	
10.5. Evaluating Transaction Results	
10.6. Transaction Management with SQL	
10.7. Transaction Log, Transaction Types	
10.8. Transaction Isolation Levels	

RECOMMENDED BOOKS

1. Fundamentals of Database Systems by Ramez Elmasri, Shamkant B. Navathe (7th Edition, 2022), Pearson Publishing, **ISBN: 978-0133970777**
2. Introduction to Database Systems by C. J. Date (8th Edition, 2003), Addison-Wesley, **ISBN: 0321197844**

INSTRUCTIONAL OBJECTIVE

1. Introduction to Database

- 1.1. Introduction to Data and Information: Understand the difference between data and information and why it's important.
- 1.2. Traditional File-Based Systems: Learn how traditional file systems work and how they store data.
- 1.3. File-Based Approach: Understand the methods used in the file-based approach to data storage.
- 1.4. Limitations of the File-Based Approach: Identify the problems and limitations of using file-based systems.
- 1.5. Introducing the Database: Discover what a database is and how it differs from file-based systems.
- 1.6. Structure of Database: Learn about the basic structure and components of a database.
- 1.7. Uses of DBMS: Explore the different uses and applications of Database Management Systems (DBMS).
- 1.8. Advantages and Disadvantages of DBMS: Understand the pros and cons of using a DBMS.
- 1.9. Components of DB Applications: Identify the key components that make up database applications.
- 1.10. Introduction to DBMS: Gain an overview of what a DBMS is and its main functions.
- 1.11. Role in Database Environment: Understand the role and importance of DBMS in managing databases.
- 1.12. DB Tools; Microsoft Access, MySQL, SQL Server, Oracle: Get introduced to popular database tools and their features.

2. Database Environment

- 2.1. Database Models Some famous database models have been introduced
- 2.2. Object-Based Data Models: Learn about data models that use objects to represent data
- 2.3. Record-Based Data Models: Understand data models that use records, like tables and rows, to represent data.
- 2.4. Physical Data Models: Explore how data is physically stored on hardware, and what is a physical data model.
- 2.5. Conceptual Modelling: Understand how to create a high-level design of a database, and what the difference between conceptual and physical database model is.
- 2.6. Network and Hierarchical Models: Learn about data models that use network and tree like structures to represent data.
- 2.7. NoSQL Graph Database: Get introduced to NoSQL databases and understand how graph databases work.

- 2.8. Database Architecture
- 2.9. Traditional Two-Tier Client–Server Architecture: Understand the basic two-layer structure of client-server databases.
- 2.10. Three-Tier Client–Server Architecture: Learn about the three-layer structure that separates the user interface, application logic, and database storage.
- 2.11. N-Tier Architectures: Explore more complex, multi-layered database architectures.
- 2.12. Database Languages
- 2.13. Data Definition Language: Learn about the language used to define the structure of database objects.
- 2.14. Data Manipulation Language: Understand the language used to manage and manipulate data in a database.
- 2.15. Data Control Language: Get familiar with the language used to control access to data in a database.
- 2.16. Introduction to Cloud-Based Databases: Understand what cloud databases are and how they are used.
- 2.17. Introduction to Distributed Databases: Learn about databases that are spread across multiple locations.

3. Relational Database Management System

- 3.1. Database Entities and Attributes: Identify and understand the basic elements of a database.
- 3.2. Database Relations & Their Types: Learn about different types of relationships between data entities.
- 3.3. Relational Keys
- 3.4. Database Schema: Understand the blueprint of a database and its structure.
- 3.5. Integrity Constraints: Learn about rules that ensure the accuracy and consistency of data.
- 3.6. Entity Integrity & Referential Integrity: Understand the importance of unique identifiers and maintaining consistent relationships.
- 3.7. Relations of Primary and Secondary Key, Cascading: Learn how primary and secondary keys are used and how cascading affects data.
- 3.8. Referencing in Database: Understand how data in one table can reference data in another table.

4. Relational Algebra and SQL

- 4.1. Relational Algebra: Learn the mathematical foundation for manipulating relational databases.
- 4.2. Built-in Functions: Understand the functions provided by SQL for common operations.

- 4.3. Unary Operations and Binary Operations: Learn about single-table and multi-table operations in SQL.
- 4.4. Set Operations: Explore operations that combine data from multiple tables.
- 4.5. Joins Operations: Understand how to combine data from two or more tables based on related columns.
- 4.6. Types of Joins: Learn the different ways to join tables (e.g., inner join, outer join).
- 4.7. Division Operation: Understand the division operation in relational algebra.
- 4.8. Aggregation and Grouping Operations: Learn how to summarize and group data in SQL.

5. Structuring Database

- 5.1. Data Definition Language: Understand how to define database structures.
- 5.2. Data Manipulation Language: Learn how to insert, update, and delete data.
- 5.3. Insertion, Deletion and Updating: Understand how to add, remove, and modify data in a database.
- 5.4. Truncation & Concatenation: Learn how to remove all records from a table and combine strings.
- 5.5. Simple Queries
- 5.6. Sorting Results (ORDER BY Clause): Learn how to sort query results.
- 5.7. Using the SQL Aggregate Functions: Understand how to perform calculations on data sets.
- 5.8. Grouping Results (GROUP BY Clause): Learn how to group data based on one or more columns.
- 5.9. Subqueries
- 5.10. SELECT Clause: Understand how to specify columns to retrieve.
- 5.11. FROM Clause: Learn how to specify the tables to query.
- 5.12. WHERE Clause: Understand how to filter query results.
- 5.13. HAVING Clause: Learn how to filter groups of data.
- 5.14. ANY and ALL: Understand how to use these operators in queries.
- 5.15. Multi-table Queries: Learn how to write queries that involve multiple tables.
- 5.16. EXISTS and NOT EXISTS: Understand how to test for the existence of data.
- 5.17. Combining Result Tables (UNION, INTERSECT, EXCEPT): Learn how to combine query results.

6. Normalization of Database Tables

- 6.1. Need for Normalization: Understand why normalization is important for database design.
- 6.2. First Normal Form: Learn how to organize data to eliminate repeating groups.
- 6.3. Second Normal Form: Understand how to eliminate redundant data.
- 6.4. Third Normal Form: Learn how to ensure data dependencies are logical.

- 6.5. Boyce-Codd Normal Form (BCNF): Understand this advanced form of normalization to handle anomalies.

7. Database Life Cycle (DBLC)

- 7.1. Database Initial Study: Learn how to start a database project by understanding requirements.
- 7.2. Database Design: Understand the steps involved in designing a database.
- 7.3. Database Design Strategies: Learn different approaches to designing databases.
- 7.4. Centralized versus Decentralized Design: Understand the differences between centralized and decentralized database designs.

8. Entity Relationship (E-R) Modelling

- 8.1. Basic Modelling Concepts: Learn the basic concepts of E-R modeling.
- 8.2. Degrees of Data Abstraction: Understand different levels of data abstraction.
- 8.3. Association and Cardinality: Learn about relationships between entities and their cardinality.
- 8.4. Relationship Participation: Understand how entities participate in relationships.
- 8.5. Composite Entities, Entity Super types and Subtypes: Learn about complex entities and their hierarchies.
- 8.6. Enhanced Entity Relationship Diagram: Explore advanced E-R modeling techniques.
- 8.7. Transform ER/EER to Relational Model: Understand how to convert E-R models into relational database schemas.

9. Views, Indexes and Transaction Management

- 9.1. Introduction to Views & Their Types: Learn what views are and their different types.
- 9.2. Introduction to Indexing: Understand what indexing is and how it improves database performance.
- 9.3. Introduction of Triggers: Learn what triggers are and how they automate tasks.
- 9.4. What is a Transaction?: Understand what database transactions are and their importance.
- 9.5. Evaluating Transaction Results: Learn how to check the results of transactions.
- 9.6. Transaction Management with SQL: Understand how to manage transactions using SQL.
- 9.7. Transaction Log, Transaction Types: Learn about the transaction log and different types of transactions.
- 9.8. Transaction Isolation Levels: Understand how to control the visibility of transactions to ensure data integrity

List of Practicals

1. Basic SQL Operations:

1. Create a database.
2. Create tables with various data types.
3. Insert data into tables
4. Update and delete records.
5. Retrieve data using SELECT statements.

2. Advanced SQL Operations:

1. Write SQL queries using WHERE, ORDER BY, and GROUP BY clauses.
2. Perform JOIN operations on multiple tables.
3. Create and use views.
4. Write subqueries and nested queries.

3. Data Manipulation and Transaction Management:

1. Perform transactions using COMMIT and ROLLBACK.
2. Use INSERT, UPDATE, and DELETE within transactions.

4. Indexing and Optimization:

1. Create indexes on tables.
2. Analyze and optimize SQL queries.

MySQL Practical Exercises:

Project 1: Online Bookstore

1. Create a database named "OnlineBookstore."
2. Design tables for "Books," "Authors," and "Customers."
3. Populate the tables with sample data.
4. Write a query to retrieve a list of books along with their authors.
5. Update the price of all books published before 2010.
6. Create a stored procedure to calculate the total sales for a given customer.

Project 2: Student Management System

1. Create a database named "StudentManagement."
2. Design tables for "Students," "Courses," and "Enrollments."
3. Populate the tables with sample data.
4. Write a query to list students who have enrolled in a specific course.
5. Insert a new student and enroll them in multiple courses in a single transaction.
6. Create a trigger to automatically update the student's GPA when new grades are added.

Project 3: Inventory Management

1. Create a database named "Inventory Management."
2. Design tables for "Products," "Suppliers," and "Orders."
3. Populate the tables with sample data.
4. Write a query to display the current stock of each product.
5. Create a stored procedure to place a new order and update the inventory accordingly.

6. Implement a view that shows the details of products running low in stock.

MS Access Project-Based Practical Exercises:

Project 1: Event Management

1. Create a database named "Event Management."
2. Design tables for "Events," "Attendees," and "Registrations."
3. Populate the tables with sample data.
4. Create a form for event registration.
5. Generate a report that lists all attendees for a specific event.
6. Implement a query to find events with the highest registration.

Project 2: Employee Information System

1. Create a database named "Employee Information."
2. Design tables for "Employees," "Departments," and "Salaries."
3. Populate the tables with sample data.
4. Build a form for entering new employee details.
5. Create a report displaying employee details, sorted by department.
6. Write a query to calculate the average salary for each department.

Project 3: Project Tracking

1. Create a database named "Project Tracking."
2. Design tables for "Projects," "Tasks," and "Employees."
3. Populate the tables with sample data.
4. Design a form for tracking task assignments.
5. Generate a report showing the progress of each project.
6. Write a query to find employees who are not assigned to any project.

CIT-293 Computer Graphic Designing

Total Contact Hours

Theory	32	T	P	C
Practical	192	1	6	3

This course offers a thorough introduction to graphic design, beginning with its fundamentals, history, and key tools, and exploring its applications in various fields, distinguishing between print and digital media. It delves into design elements such as line, shape, form, space, and texture, along with principles like balance, proximity, alignment, contrast, and symmetry. The curriculum covers color theory, typography, and hands-on training with Adobe Photoshop and Illustrator. The course concludes with an exploration of brand identity and logo design, including the importance of branding and creating various marketing materials.

Learning Outcomes: The students will be able to

- Understand basic concepts to graphic design, principle and elements of design
- Understand the concept of color families, color wheels, primary, secondary and tertiary colors.
- Understand fonts, font families, font types and alignment.
- Implement design concepts in Adobe Photoshop, illustrator and Canva.
- Design banners, brochures, flyers, pamphlets, billboards and packages
- Create logos keeping in view the types of logos.

COURSE OUTLINE

Contents	Hours
1. Introduction to Graphic Design	03 Hrs
1.1. Graphic design	
1.2. Brief History of Graphic Design	
1.3. Overview of design software and tools	
1.4. Application of graphic design in different field	
1.5. Difference between Print Media and Digital Media	
2. Design Elements & Principal	04 Hrs
2.1. Elements of design	
2.1.1. Line (Divide, Connect, Hierarchy, Decoration)	
2.1.2. Shape (Geometric, Organic, Negative)	
2.1.3. Form	
2.1.4. Space (White, Negative, Depth)	
2.1.5. Texture (Image, Surface, Pattern)	

- 2.2. Principle of Design
 - 2.2.1. Balance
 - 2.2.2. Proximity
 - 2.2.3. Alignment
 - 2.2.4. Contrast
 - 2.2.5. Symmetric Design & Asymmetric Design

3. Color Theory **04 Hrs**

- 3.1. Color Wheel
 - 3.1.1. Primary Colors
 - 3.1.2. Secondary colors
 - 3.1.3. Tertiary Colors
- 3.2. Color Schemes & Harmony
 - 3.2.1. Monochromatic
 - 3.2.2. Analogous
 - 3.2.3. Complimentary
 - 3.2.4. Split Complimentary
 - 3.2.5. Triadic
 - 3.2.6. Tint
 - 3.2.7. Shade
 - 3.2.8. Tone
- 3.3. Color Models: CMYK, RGB, HSB etc

4. Typography **03 Hrs**

- 4.1. Fonts
- 4.2. Types of Fonts
- 4.3. Alignment of Text
- 4.4. Leading
- 4.5. Character Spacing
- 4.6. Kerning
- 4.7. Widow Orphan
- 4.8. Tracking

5. Adobe Photoshop **07 Hrs**

- 5.1. Raster Image & Canvas
- 5.2. Raster Files
- 5.3. Selection Tools
- 5.4. Lasso, Polygon Lasso, Magnetic Lasso
- 5.5. Magic wand, Quick Selection
- 5.6. Layers in Photoshop
- 5.7. Layer Styles

- 5.8. Clone Stamp
- 5.9. Healing, Patch, Content Aware Move
- 5.10. Painting Tools
- 5.11. Filters
- 5.12. Animation

6. **Adobe Illustrator**

07 Hrs

- 6.1. Vector Image & Art board
- 6.2. Vector Files
- 6.3. Selection & Direct Selection
- 6.4. Color, Swatches
- 6.5. Vector Shapes
- 6.6. Working with Layers
- 6.7. Pen Tool
- 6.8. Pathfinder
- 6.9. Scale & Rotate Tool
- 6.10. Gradient Tool

7. **Brand Identity and Logo Design**

04 Hrs

- 7.1. Importance of branding
- 7.2. Logo & Types of Logo
- 7.3. Office Stationary
- 7.4. Banner, Flyers, Pamphlet, Brochure, Billboards
- 7.5. Packaging

RECOMMENDED BOOKS

- 1. The Visual Dictionary of Graphic Design Gavin Ambrose and Paul Haris
- 2. The Fundamentals of Graphic Design Gavin Ambrose and Paul Haris
- 3. Adobe Photoshop Classroom in a Book By Conrad Chavez
- 4. Adobe Illustrator Classroom in a Book By Brian Wood

INSTRUCTIONAL OBJECTIVES

1. Introduction to Graphic Design

- 1.1. Define graphic design, its importance, and its role in visual communication.
- 1.2. Explain historical evolution of graphic design, Cave Paintings, Egyptian Civilization (1st Generation), Hieroglyphs, invention of paper in china, printing in Europe, first decorative firm and first advertising agency
- 1.3. Describe the key software and tools used in graphic design. Adobe family, CANVA and FIGMA
- 1.4. Explain various fields where graphic design is applied and explain its significance in each field
- 1.5. Distinguish between print and digital media, including their characteristics and applications.

2. Design Elements & Principal

- 2.1. Explain Elements of design
 - 2.1.1. Explain importance of lines in design including connecting, divide, decoration and hierarchy.
 - 2.1.2. Differentiate between geometric, organic and negative shapes.
 - 2.1.3. Explain the concept of form and its use in creating three dimensional effects.
 - 2.1.4. Explain the use and importance of white, negative space and depth.
 - 2.1.5. Explain texture and differentiate between image, surface and pattern texture.
- 2.2. Explain Principle of Design
 - 2.2.1. Define Balance with example.
 - 2.2.2. Define Proximity with example.
 - 2.2.3. Explain different alignments, left, right center and justify.
 - 2.2.4. Define Contrast with example.
 - 2.2.5. Define Symmetric & Asymmetric Design with examples

3. Color Theory

- 3.1. Define Color Wheel of RGB, CMYK and RYB
 - 3.1.1. Explain Primary Colors and their role in color mixing.
 - 3.1.2. Explain Secondary colors and their creation from primary colors.
 - 3.1.3. How Tertiary Colors are formed by mixing primary and secondary colors.
- 3.2. Explain Color Schemes and color associations
 - 3.2.1. Explain Monochromatic schemes and how they are formed.
 - 3.2.2. Explain Analogous schemes and how they are created from color wheels.
 - 3.2.3. Explain Complimentary schemes and how they are created from color wheels.
 - 3.2.4. Explain Split Complimentary schemes and how they are created from color wheels
 - 3.2.5. Explain Triadic schemes and how they are created from color wheels
 - 3.2.6. Define Tint
 - 3.2.7. Define Shade
 - 3.2.8. Define Tone
- 3.3. Explain and differentiate between CMYK, RGB and HSB colors.

4. Typography

- 4.1. Explain different font styles and their usage.
- 4.2. Differentiate between Serif, Sans-Serif, Script and decorative fonts and where they are used.
- 4.3. Explain vertical and horizontal alignments. Explain nine alignments like top left, top center etc.
- 4.4. Define Leading and its importance in reading.
- 4.5. Define Character Spacing and its effect on visual appeal.
- 4.6. Define Kerning and why fonts with bad kerning are not recommended.
- 4.7. Define Widow and Orphan in document
- 4.8. Define Tracking

5. Adobe Photoshop

- 5.1. Explain raster images. What is canvas?
- 5.2. Explain raster files and extensions of raster files.
- 5.3. Define Selection Tools
- 5.4. Define Lasso, Polygon Lasso, Magnetic Lasso
- 5.5. Define Magic wand, Quick Selection
- 5.6. Explain layers in Photoshop and its importance while creating .psd files
- 5.7. Explain Layer Styles. Shadows, Glow, Gradient etc.
- 5.8. What is Clone Stamp and where it is used.
- 5.9. Define Healing, Patch, Content Aware Move.
- 5.10. Define Painting Tools.
- 5.11. Define basic filters
- 5.12. Define animation and its usage.

6. Adobe Illustrator

- 6.1. Explain vector images. What is Art-board?
- 6.2. Explain vector files and extensions of vector files.
- 6.3. Define Selection & Direct Selection tools and differentiate between them
- 6.4. Explain Color spectrum, Swatches, and color guide.
- 6.5. Define vector shapes e.g. rectangle, polygon, hexagon etc.
- 6.6. How layers work and how they are different from Photoshop.
- 6.7. Define Pen Tool and its use in tracing.
- 6.8. Define Pathfinder
- 6.9. Define Scale & Rotate Tool
- 6.10. Define Gradient and basic difference from normal colors

7. Brand Identity and Logo Design

- 7.1. Explain importance of branding in graphics
- 7.2. Define and explain seven types of logos. Monogram/Lettermark, Word Mark, Pictorial, Abstract Mark, Mascot, Combination and Emblem Logo

- 7.3. Explain Office Stationary e.g. letterhead, envelop, business card and DVD cover.
- 7.4. Define Banner, Flyers, Pamphlet, Brochure and Billboards sizes.
- 7.5. Define Packaging and its importance.

LIST OF PRACTICAL

Adobe Photoshop

1. Software Installation & orientation, Move tool and Marquee Tool
2. Lasso, Polygon Lasso and Magnetic Lasso Tool
3. Quick Selection and Magic Wand Tool, Pen Tool
4. Create Typographical Text
5. Change mode of file from CMYK to RGB & vice versa Crop Tool, Eyedropper and Paint Bucket
6. Spot Healing, Healing Brush and Patch Tool
7. Background Layer, Add/Delete/Merge Layers
8. Clone Stamp Tool & pattern stamp tool
9. Removing scratches and restoring old photo.
10. Editing Face beauty using Photo tools (removing moles, pimples and wrinkles)
11. Use different Shapes in Photoshop (Rounded, Line, Ellipse etc)
12. Layer Mask, Clipping Mask,
13. Different features of layer style
14. Smudge and Sponge Tool
15. Remove background from Picture and replace the background. Crop the image to passport size.
16. Remove background from Logos and save as PNG file.
17. Use pen tool to extract any object from the picture.
18. Use Mask to create and mix image (Color and Black & White) levels, hue and saturation, curves
19. Animation using Timeline
20. Make an animated advertisement of any Famous brand.
21. Animate the object in Photoshop. Save the file as animated gif.

Adobe Illustrator

22. Software Installation & Orientation Selection Tool, Direct Selection & Pen Tool.
23. Rotate Tool & Transform Tool.
24. Type Tool: Character Formatting, Paragraph Formatting.
25. Path Finder Tool: Unite, Minus Front, Intersect, Trim and Divide.
26. Color Pattern and Swatches Tool
27. Gradient Tool
28. Design Visiting Card
29. Design Letter Head
30. Design office stationary (Visiting card, letterhead, envelop, file folder)

CANVA

31. Introduction to CANVA.
32. Creates shapes and write text
33. Design banner and pamphlets in CANVA

CIT- 204 Web Development

Total Contact Hours

Theory	64	T	P	C
Practical	192	2	6	4

This web development course provides a comprehensive overview of the key technologies and concepts in web development. It covers the fundamentals of web applications, front-end development (HTML, CSS, JavaScript, React, Angular), back-end development (Node Js, Python Flask), and database management (MongoDB). Additionally, it introduces students to popular content management systems (CMS) WordPress

Learning Outcomes: The students will be able to

- Gain a thorough understanding of web technologies and Implement web applications from scratch.
- Develop responsive front-end interfaces using HTML, CSS, MongoDB, JavaScript and its frameworks React and Angular, Python and Python-based framework Flask.
- Build scalable back-end systems and websites using Node.js and CMS platforms like WordPress.
- Acquire the skills necessary to independently develop, robust web solutions.

COURSE OUTLINE

Contents	Hours
1. HTML Basics	04 Hrs
1.1. Introduction to HTML	
1.1.1.Overview of HTML and its significance in web development	
1.1.2.Structure of an HTML document	
1.2. Basic HTML Tags	
1.2.1.Headings, paragraphs, lists	
1.2.2.Emphasis and strong tags	
1.2.3.Div and span elements	
1.3. Table Management	
1.3.1.Creating tables	
1.3.2.Table rows, cells, headers	
1.3.3.Merging cells (colspan and rowspan)	
1.4. Forms	
1.4.1.Form elements: input, radio, checkbox, select, text area	
1.4.2.Form attributes and validation	
1.4.3.Handling form submission	

1.5. Images and Links Handling

- 1.5.1. Inserting and formatting images
- 1.5.2. Creating hyperlinks

2. CSS Fundamentals

04 Hrs

2.1. Introduction to CSS

- 2.1.1. What is CSS and its role in web development
- 2.1.2. How to include CSS in HTML

2.2. CSS Selectors

- 2.2.1. Element, class, and ID selectors
- 2.2.2. Attribute selectors and pseudo-classes
- 2.2.3. Specificity and inheritance

2.3. Color Management

- 2.3.1. Setting colors using names, hex, RGB, and HSL
- 2.3.2. Background colors and gradients

2.4. Margin and Padding Management

- 2.4.1. Box model overview
- 2.4.2. Margin and padding properties

2.5. Layouts

- 2.5.1. Display property (block, inline, inline-block)
- 2.5.2. Positioning (static, relative, absolute, fixed)
- 2.5.3. Flexbox and Grid systems

2.6. Creating a Simple Static Website

- 2.6.1. Linking multiple pages
- 2.6.2. Building a simple e-commerce website layout

3. Bootstrap Framework

06 Hrs

3.1. Introduction to Bootstrap

- 3.1.1. Overview of Bootstrap and its benefits
- 3.1.2. How to include Bootstrap in your project

3.2. Basic Classes

- 3.2.1. Typography and text utilities
- 3.2.2. Buttons, alerts, and badges

3.3. Forms and Tables

- 3.3.1. Bootstrap form classes
- 3.3.2. Styling tables with Bootstrap

3.4. Bootstrap Grid System

- 3.4.1. Understanding the grid system
- 3.4.2. Creating responsive layouts

3.5. Responsive Design

- 3.5.1. Bootstrap for mobile and desktop views
- 3.5.2. Converting the e-commerce website to use Bootstrap

4. JavaScript Fundamentals	08 Hrs
4.1. Introduction to JavaScript	
4.1.1.Importance of JavaScript in web development	
4.1.2.Including JavaScript in HTML	
4.2. Variables and Operators	
4.2.1.Declaring variables	
4.2.2.Using assignment operators	
4.3. Control Structures	
4.3.1.If conditions and else statements	
4.3.2.Switch cases	
4.4. Loops and Functions	
4.4.1.For, while, and do-while loops	
4.4.2.Defining and invoking functions	
4.5. Basic Events in JavaScript	
4.5.1.Event handling	
4.5.2.DOM manipulation	
4.6. Form Validation	
4.6.1.Validating form input using JavaScript	
4.6.2.Creating a calculator application	
5. jQuery	02 Hrs
5.1. Introduction to jQuery	
5.1.1.Overview of jQuery and its advantages	
5.1.2.Including jQuery in your project	
5.2. Basic jQuery Operations	
5.2.1.Selecting elements	
5.2.2.Manipulating the DOM	
5.3. Simple Animations	
5.3.1.Fading, sliding, and other basic animations	
5.3.2.Custom animations using jQuery	
5.4. Adding and Removing HTML	
5.4.1.Dynamically adding content	
5.4.2.Removing elements from the DOM	
6. Introduction to Databases	02 Hrs
6.1. Introduction to MongoDB	
6.1.1.Overview of MongoDB	
6.1.2.Installing and setting up MongoDB	
6.1.3.Simple CRUD operations	
7. JavaScript as Backend	14 Hrs
7.1. Node.js Installation and Setup	
7.1.1.Installing Node.js	
7.1.2.Setting up a Node.js project	
7.2. Creating a Simple Express.js	
7.2.1.Writing a basic server	

7.2.2. Handling requests and responses	
7.3. Building APIs with Express.js	
7.3.1. Callback functions	
7.3.2. Creating your first API	
7.3.3. Using the API in front-end applications	
7.4. Project Structure (Model, Controller, Router)	
7.4.1. Organizing controllers, routers, models, and utilities	
7.4.2. Best practices for project structure	
7.5. Advanced API Development	
7.5.1. Creating multiple APIs	
7.5.2. Consuming APIs from the front-end	
7.5.3. Storing data in the database via APIs	
7.6. Authentication and Authorization	
7.6.1. Implementing login functionality	
7.6.2. Handling authentication and authorization	
7.7. E-commerce Cart Application	
7.7.1. Building an e-commerce cart	
7.7.2. Viewing order details in the dashboard	
7.8. Middleware and Order Management	
7.8.1. Using middleware in APIs	
7.8.2. Managing and viewing orders	
7.9. Postman and API Testing	
7.9.1. Introduction to Postman	
7.9.2. Setting up and using Postman	
7.9.3. Testing APIs with Postman	
8. JavaScript based Frontend Frameworks	04 Hrs
8.1. Introduction to Frontend Frameworks	
8.1.1. Overview of React and Angular	
8.2. Use Cases and Applications	
8.2.1. Building simple applications with each framework	
9. Web Development with Python	12 Hrs
9.1. Introduction to Python Programming	
9.1.1. Data Types: Numbers, strings, lists, tuples, dictionaries	
9.1.2. Operators: Arithmetic, comparison, logical, assignment	
9.1.3. Control Flow: Conditional statements (if, else, elif), loops (for, while)	
9.1.4. Functions: Defining, calling, parameters, return values	
9.1.5. Modules and Packages: Importing modules, creating custom modules	
9.2. Introduction to Flask	
9.2.1. What is a web framework?	
9.2.2. Flask's role in web development.	
9.2.3. Core components of a Flask application.	
9.3. Setting Up a Flask Environment:	
9.3.1. Installing Flask and required dependencies.	
9.3.2. Creating a basic Flask application.	

9.3.3. Running the application.	
9.4. Routes and Request Handling:	
9.4.1. Defining routes using Flask's @app.route decorator.	
9.4.2. Handling different HTTP methods (GET, POST, etc.).	
9.4.3. Accessing request data (query parameters, form data, JSON).	
9.5. Rendering Templates	
9.5.1. Introduction to template engines.	
9.5.2. Using Jinja2 with Flask.	
9.5.3. Creating dynamic HTML templates.	
9.5.4. Passing data from Python to templates.	
9.6. Forms	
9.6.1. Creating HTML forms.	
9.6.2. Handling form submissions using Flask-WTF.	
9.6.3. Validating form data.	
9.7. Databases with Flask-SQLAlchemy	
9.7.1. Introduction to databases and ORMs.	
9.7.2. Using Flask-SQLAlchemy to interact with databases.	
9.7.3. Creating models and database tables.	
9.7.4. Performing CRUD operations.	
9.8. Error Handling	
9.8.1. Handling exceptions in Flask applications.	
9.8.2. Custom error pages.	
9.9. Deployment	
9.9.1. Basic understanding of deployment options.	
9.9.2. Deploying Flask apps to platforms like Heroku or PythonAnywhere.	
10. DNS, Nameservers, and Hosting	02 Hrs
10.1. Understanding DNS and Nameservers	
10.1.1. What is DNS	
10.1.2. How nameservers work	
10.2. Types of Hosting	
10.2.1. Shared hosting, VPS, dedicated servers	
10.2.2. Cloud hosting overview	
10.3. Setting Up Hosting	
10.3.1. Deploying projects to different servers	
10.3.2. Configuring hosting environments	
11. WordPress CMS	06 Hrs
11.1. Introduction to WordPress	
11.1.1. What is WordPress	
11.1.2. Benefits of using WordPress	
11.2. Installation and Setup	
11.2.1. Installing WordPress	
11.2.2. Setting up a basic website	
11.3. Themes and Plugins	
11.3.1. Choosing and customizing themes	

11.3.2. Installing and using plugins

11.4. WooCommerce

11.4.1. Setting up an e-commerce store with WooCommerce

11.4.2. Managing products and orders

RECOMMENDED BOOK

- HTML and CSS: Design and Build Websites, by Duckett, Jon, 2011, John Wiley & Sons, ISBN: 978-1118008188
- JavaScript and jQuery: Interactive Front-End Web Development by Duckett, Jon, 2014, John Wiley & Sons, ISBN: 978-1118531648
- Node.js Web Development - Third Edition: Create real-time server-side applications with this practical, step-by-step guide, 2016, Packt Publishing, ISBN: 978-1785881503
- MongoDB: The Definitive Guide: Powerful and Scalable Data Storage 3rd Edition by Chodorow, Kristina, 2019, O'Reilly Media ISBN: 978-1491954461
- The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations by Gene Kim, Jez Humble, Patrick Debois, and John Willis, 2016, IT Revolution Press, ISBN: 978-1942788003
- Flask Web Development: Developing Web Applications with Python (2nd Ed.) by Miguel Grinberg, O'Reilly Media (2018), ISBN: 978-1491991732

INSTRUCTIONAL OBJECTIVES

1. HTML Basics

1.1. Introduction to HTML

- 1.1.1. Understand what HTML is and why it is important for web development.
- 1.1.2. Learn the basic structure of an HTML document.

1.2. Basic HTML Tags

- 1.2.1. Use headings, paragraphs, and lists to organize content.
- 1.2.2. Apply Emphasis and Strong tags to highlight text.
- 1.2.3. Utilize div and span elements to structure HTML content.

1.3. Table Management

- 1.3.1. Create and format tables in HTML.
- 1.3.2. Use table rows, cells, and headers effectively.
- 1.3.3. Merge cells using colspan and rowspan attributes.

1.4. Forms

- 1.4.1. Identify and use different form elements such as input, radio, checkbox, select, and text area.
- 1.4.2. Understand form attributes and basic validation.
- 1.4.3. Handle form submissions.

1.5. Images and Links Handling

- 1.5.1. Insert and format images in an HTML document.
- 1.5.2. Create hyperlinks to navigate between web pages.

2. CSS Fundamentals

2.1. Introduction to CSS

- 2.1.1. Explain what CSS is and its role in web development.
- 2.1.2. Include CSS in an HTML document using various methods.

2.2. CSS Selectors

- 2.2.1. Use element, class, and ID selectors to apply styles.
- 2.2.2. Utilize attribute selectors and pseudo-classes.
- 2.2.3. Understand specificity and inheritance in CSS.

2.3. Color Management

- 2.3.1. Set colors using names, hex, RGB, and HSL values.
- 2.3.2. Apply background colors and gradients.

2.4. Margin and Padding Management

- 2.4.1. Understand the box model.
- 2.4.2. Apply margin and padding properties.

2.5. Layouts

- 2.5.1. Use the display property (block, inline, inline-block) to control layout.
- 2.5.2. Position elements using static, relative, absolute, and fixed positioning.
- 2.5.3. Implement Flexbox and Grid systems for layout design.

2.6. Creating a Simple Static Website

- 2.6.1. Link multiple HTML pages together.
- 2.6.2. Build a simple layout for an e-commerce website.

3. Bootstrap Framework

3.1. Introduction to Bootstrap

- 3.1.1. Understand what Bootstrap is and its benefits.
- 3.1.2. Include Bootstrap in a web project.

3.2. Basic Classes

- 3.2.1. Use Bootstrap classes for typography and text utilities.
- 3.2.2. Implement buttons, alerts, and badges using Bootstrap.

3.3. Forms and Tables

- 3.3.1. Style forms using Bootstrap form classes.
- 3.3.2. Apply Bootstrap styles to tables.

3.4. Bootstrap Grid System

- 3.4.1. Understand and use the Bootstrap grid system.
- 3.4.2. Create responsive layouts with Bootstrap.

3.5. Responsive Design

- 3.5.1. Use Bootstrap to create layouts for mobile and desktop views.
- 3.5.2. Convert a static e-commerce website to use Bootstrap for responsiveness.

4. JavaScript Fundamentals

4.1. Introduction to JavaScript

- 4.1.1. Explain the importance of JavaScript in web development.
- 4.1.2. Include JavaScript in an HTML document.

4.2. Variables and Operators

- 4.2.1. Declare and use variables in JavaScript.
- 4.2.2. Understand and use assignment operators.

4.3. Control Structures

- 4.3.1. Implement if conditions and else statements.
- 4.3.2. Use switch cases for decision making.

4.4. Loops and Functions

- 4.4.1. Use for, while, and do-while loops for iteration.
- 4.4.2. Define and invoke functions in JavaScript.

4.5. Basic Events in JavaScript

- 4.5.1. Handle events in JavaScript.
- 4.5.2. Manipulate the DOM using JavaScript.

4.6. Form Validation

- 4.6.1. Validate form inputs using JavaScript.
- 4.6.2. Create a simple calculator application.

5. JQuery

5.1. Introduction to jQuery

- 5.1.1. Understand what jQuery is and its advantages.
- 5.1.2. Include jQuery in a web project.

5.2. Basic jQuery Operations

- 5.2.1. Select elements using jQuery.
- 5.2.2. Manipulate the DOM with jQuery.

5.3. Simple Animations

- 5.3.1. Create basic animations like fading and sliding.

- 5.3.2. Implement custom animations using jQuery.
- 5.4. Adding and Removing HTML**
 - 5.4.1. Dynamically add content to the DOM.
 - 5.4.2. Remove elements from the DOM.
- 6. Introduction to Databases**
 - 6.1. Introduction to MongoDB**
 - 6.1.1. Get an overview of MongoDB.
 - 6.1.2. Install and set up MongoDB.
 - 6.1.3. Perform simple CRUD operations with MongoDB.
- 7. JavaScript as Backend**
 - 7.1. Node.js Installation and Setup**
 - 7.1.1. Install Node.js on your system.
 - 7.1.2. Set up a Node.js project.
 - 7.2. Creating a Simple Express.js**
 - 7.2.1. Write a basic server using Express.js.
 - 7.2.2. Handle requests and responses.
 - 7.3. Building APIs with Express.js**
 - 7.3.1. Understand and use callback functions.
 - 7.3.2. Create your first API.
 - 7.3.3. Use the API in front-end applications.
 - 7.4. Project Structure (Model, Controller, Router)**
 - 7.4.1. Organize controllers, routers, models, and utilities.
 - 7.4.2. Follow best practices for project structure.
 - 7.5. Advanced API Development**
 - 7.5.1. Create multiple APIs.
 - 7.5.2. Consume APIs from the front-end.
 - 7.5.3. Store data in the database via APIs.
 - 7.6. Authentication and Authorization**
 - 7.6.1. Implement login functionality.
 - 7.6.2. Handle authentication and authorization.
 - 7.7. E-commerce Cart Application**
 - 7.7.1. Build an e-commerce cart.
 - 7.7.2. View order details in the dashboard.
 - 7.8. Middleware and Order Management**
 - 7.8.1. Use middleware in APIs.
 - 7.8.2. Manage and view orders.
 - 7.9. Postman and API Testing**
 - 7.9.1. Introduction to Postman.
 - 7.9.2. Set up and use Postman.
 - 7.9.3. Test APIs with Postman.
- 8. Frontend Frameworks**
 - 8.1. Introduction to Frameworks**
 - 8.1.1. Get an overview of React, Angular

8.2. Use Cases and Applications

8.2.1. Build simple applications using each framework.

9. Web Development with Python

9.1. Introduction to Python Programming

9.1.1. Understand the basics of Python programming, including data types, variables, and control flow.

9.1.2. Learn how to write simple Python programs.

9.2. Introduction to Flask

9.2.1. Understand the concept of web frameworks.

9.2.2. Learn about Flask as a Python web framework.

9.2.3. Know the basic structure of a Flask application.

9.3. Setting Up a Flask Environment

9.3.1. Set up a Python environment for Flask development.

9.3.2. Install Flask and necessary dependencies.

9.3.3. Create a basic Flask application.

9.4. Routes and Request Handling

9.4.1. Understand how to define URL routes in a Flask app.

9.4.2. Learn how to handle different types of requests (GET, POST, etc.).

9.4.3. Process incoming data from requests.

9.5. Rendering Templates

9.5.1. Use templates to create dynamic web pages.

9.5.2. Learn how to pass data from Python to templates.

9.5.3. Create reusable templates.

9.6. Forms

9.6.1. Build HTML forms using Flask-WTF.

9.6.2. Validate user input from forms.

9.6.3. Handle form submissions.

9.7. Databases with Flask-SQLAlchemy

9.7.1. Understand the concept of databases and ORMs.

9.7.2. Use Flask-SQLAlchemy to interact with databases.

9.7.3. Create database models and perform CRUD operations.

9.8. Error Handling

9.8.1. Handle errors gracefully in Flask applications.

9.8.2. Create custom error pages.

9.9. Deployment

9.9.1. Understand the process of deploying a Flask app to a web server.

9.9.2. Learn about different deployment options.

9.9.3. Deploy a simple Flask app to a platform like Heroku or PythonAnywhere.

10. DNS, Nameservers, and Hosting

10.1. Understanding DNS and Nameservers

10.1.1. Understand what DNS is.

10.1.2. Learn how nameservers work.

10.2. Types of Hosting

10.2.1. Differentiate between shared hosting, VPS, and dedicated servers.

10.2.2. Get an overview of cloud hosting.

10.3. Setting Up Hosting

10.3.1. Deploy projects to different servers.

10.3.2. Configure hosting environments.

11. WordPress CMS

11.1. Introduction to WordPress

11.1.1. Understand what WordPress is and its benefits.

11.2. Installation and Setup

11.2.1. Install WordPress.

11.2.2. Set up a basic WordPress website.

11.3. Themes and Plugins

11.3.1. Choose and customize WordPress themes.

11.3.2. Install and use WordPress plugins.

11.4. WooCommerce

11.4.1. Set up an e-commerce store using WooCommerce.

11.4.2. Manage products and orders with WooCommerce.

List of Practical

HTML Practical's

1. Create a Basic HTML Page: Design a simple HTML page with headings, paragraphs, and lists.
2. Build a Table: Create a table with merged cells and headers.
3. Design a Form: Develop a form with various input elements and validation attributes.
4. Insert Images and Links: Add images and create hyperlinks in an HTML page.

CSS Practical's

1. Apply CSS Selectors: Use element, class, and ID selectors to style HTML elements.
2. Color Management with CSS: Set background colors and gradients using different color values.
3. Box Model Practice: Adjust margins and padding on various elements.
4. Flexbox Layout: Create a flexible layout using Flexbox.
5. Build a Static E-commerce Website: Link multiple pages to create a simple static e-commerce site.

Bootstrap Practical's

1. Bootstrap Typography and Buttons: Style text and buttons using Bootstrap classes.
2. Bootstrap Forms and Tables: Apply Bootstrap classes to style forms and tables.
3. Bootstrap Grid System: Create a responsive layout using Bootstrap's grid system.
4. Responsive Design Conversion: Convert a static e-commerce website to a responsive one using Bootstrap.

JavaScript Practical's

1. JavaScript Variables and Operators: Write a script that uses variables and operators.
2. Conditional Statements: Implement if conditions and switch cases in JavaScript.
3. Loops in JavaScript: Write a script using for, while, and do-while loops.
4. JavaScript Functions: Define and invoke JavaScript functions.
5. Event Handling: Add event listeners to HTML elements and handle events.
6. Form Validation with JavaScript: Validate form input using JavaScript.
7. Create a Calculator: Develop a simple calculator application using JavaScript.

JQuery Practical's

1. Basic jQuery Operations: Use jQuery to select and manipulate HTML elements.
2. jQuery Animations: Implement fading and sliding animations with jQuery.
3. Dynamic HTML with jQuery: Add and remove HTML content dynamically using jQuery.

MongoDB Practical's

1. MongoDB CRUD Operations: Perform create, read, update, and delete operations using MongoDB.

Node.js Practical's

1. Set Up Node.js: Install Node.js and create a basic project.
2. Simple Node Server: Write a simple server using Node.js.
3. Create an API: Develop your first API with Node.js.
4. API Consumption: Use a front-end application to consume a Node.js API.
5. Authentication Implementation: Add login functionality to a Node.js application.
6. API Testing with Postman: Use Postman to test your Node.js APIs.

React Practical's

Objective: Build a simple to-do list application using React.

Features:

1. Create a component to represent a single to-do item.
2. Maintain a list of to-do items in the application state.
3. Implement functionality to add new to-do items.
4. Allow users to mark to-do items as completed.
5. Provide an option to delete completed to-do items.

Deployment Practical's

1. Deploy a Project: Deploy a web project to a development, staging, or production server.

WordPress Practical's

1. Install WordPress: Install WordPress and set up a basic website.
2. Customize a WordPress Theme: Choose and customize a WordPress theme.
3. Add Plugins to WordPress: Install and configure plugins in WordPress.
4. WooCommerce Setup: Set up an e-commerce store using WooCommerce in WordPress.

Projects

Project 1(React or Angular) : Ecommerce Web Site. Online store with product listing and shopping cart and checkout system.

1. Static product catalog with basic product information (name, description, price, image).
2. Shopping cart functionality (adding, removing, viewing items).
3. Checkout process (order summary, simulation of payment gateway integration).

Project 2 (Python Flask): Blog website. Where user can create account write edit and delete posts.

1. Build a basic blog application using Flask.

2. Include features for creating, editing, and deleting posts.
3. Allow users to add comments to posts.
4. Implement user authentication and authorization.
5. Use a template for creating blog post layouts.

Project 3 (Wordpress): Development of Content Management System (CMS) Application. Create an application where user can create account, write, edit and delete contents for any application domain.
Objectives:

1. Understand the core concepts of a Content Management System (CMS).
2. Create and manage various content types within a CMS environment.
3. Utilize basic CMS features for customization and optimization.
4. Experience the workflow of publishing content on a website.

Functionality:

1. Content Management

1. Create and edit blog posts with titles, content, date and time, and featured images.
2. Organize posts into categories and tags (better to use word cloud).
3. Manage pages (About Us, Contact Us, etc.) with static content.

2. User Management

1. Create different user roles (administrator, editor, author, subscriber).
2. Assign permissions to manage content and website settings.

3. Media Library

1. Upload and manage images, videos, and other media files.
2. Insert media (images, videos, links) into posts and pages.

4. Design Customization

1. Choose and customize a pre-designed template theme.
2. Modify website layout, colors, and fonts.
3. Create custom menus for navigation.

5. Basic Analytics

1. Track website traffic and user behavior using built-in analytics tools.

سال سوئم

حصہ اول اسلامیات

تدریس مقاصد

قرآن حکیم:

عمومی مقاصد: منتخب سورتوں اور آیات کی روشنی میں اسلام کے بنیادی مقاصد اور عبادات جان سکے
 خصوصی مقاصد: طالب علم اس قابل ہو جائے گا کہ۔
 سورۃ الفاتحہ، آیت لکری، سورۃ بقرہ کی آخری آیات ازمن الرسول سے اور سورۃ اخلاق کا ترجمہ و تشریح کر سکے
 طالب علم درج ذیل کا مفہوم بیان کر سکے

- () رب العالمین صرف اللہ تعالیٰ ہے
- () اللہ رحم کرنے والا ہے
- () قیامت کے دن بادشاہی اللہ کی ہوگی
- () عبادات اور استعانت کا حقدار صرف اللہ ہے
- () طالب علم درج ذیل کا مفہوم بیان کر سکے
- () اللہ پاک ہر عیب سے پاک ہے
- () اللہ کے اسمائے حسنہ حق اور قیوم ہیں
- () تمام انبیاء پر ایمان لانا ضروری ہے
- () رسول، ملائکہ، کتب، سماویہ، پر ایمان لانا فرض ہے
- () اطاعت حقیقی صرف اللہ کے لیے ہے
- () اسلامی احکامات پر عمل کرنا انسانی بساط میں ہے
- () کفر کو اللہ کی مدد کے بغیر شکست نہیں دی جاسکتی
- () اللہ ایک ہے
- () اللہ کسی کا محتاج نہیں نہ اس کا کوئی شریک ہے

منتخب احادیث:

عمومی مقصد: احادیث کی روشنی میں اسلامی تعلیمات پر عمل پیرا ہو سکے

() اسلام کی اعلیٰ اقدار کو اپنا کر مثالی معاشرہ پیدا کر سکے

مطالعہ پاکستان (حصہ دوم)
قیام پاکستان

تدریس مقاصد:

عمومی مقاصد:

() قیام پاکستان کے بعد درپیش مسائل سے آگاہی حاصل کرے اور بیان کرے

خصوصی مقاصد:

- () باؤنڈری کمیشن تشکیل اور اس کے فرائض بیان کر سکے
- () ریڈ کلف اور اس کے ایوارڈ کے بارے میں بیان کر سکے
- () بنگال اور کلکتہ کی تقسیم کی وجوہات بیان کر سکے
- () پنجاب کی تقسیم کی تفصیل بیان کر سکے
- () مہاجرین کی آمد سے جو مسائل پیدا ہوئے انہیں بیان کر سکے
- () ریاستوں کے الحاق کے بارے میں تفصیل بیان کر سکے
- () ریاست جموں کشمیر کے بارے میں بیان کر سکے
- () نہری پانی کے تنازعہ کو بیان کر سکے
- () قرارداد مقاصد کی تفصیلات بیان کر سکے
- () 22 علماء کے متفقہ اسلامی نکات بیان کر سکے
- () 1973 کے آئین میں انسانی حقوق کی دفعات (28-8)۔ اقلیتوں کے حقوق اور دیگر محروم طبقات کے حقوق کے بارے میں جان سکے
- () قیام پاکستان کے بعد نفاذ اسلام کی کوششوں کو بیان کر سکے
- () پاکستان کے محل وقوع اور اس کی جغرافیائی اہمیت بیان کر سکے اور GSP+ کے تعارف اور عملی طریقہ کار بیان کر سکے
- () پاکستان میں قدرتی وسائل (تیل، گیس، کوئلہ) کے بارے میں بیان کر سکے

ٹی پی سی
1 0 1
کل وقت 20 گھنٹے

غیر مسلم طلباء کے لیے
GEN301

نصاب اخلاقیات
سال سوئم

موضوعات:

- () احساس ذمہ داری
- () مثبت ذہن
- () عدل و انصاف
- () قومی خدمت کا جذبہ
- () ذکر و نظر کی پاکیزگی
- () احترام آدمیت
- () شائستگی
- () عفو و درگزر
- () بردباری
- () خود انحصاری
- () اثر و نفوذ
- () جامعیت
- () اپنی ذات کی معرفت (بذریعہ ہم عمر طلباء، اساتذہ، اہم شخصیات)
- () بچوں کے حقوق
- () عورتوں کے حقوق
- () بزرگ شہریوں کے حقوق
- () محنت حضرات کے حقوق

غیر مسلم طلباء کے لیے
نصاب اخلاقیات (سال سوئم)

تدریس مقاصد:

عمومی مقاصد:

- () ملکی ترقی کے لیے اعلیٰ اوصاف کے ساتھ بہتر طور پر ملک و ملت کی خدمت کر سکے
- خصوصی مقاصد: طالب علم اس قابل ہو سکے کہ
- () موضوعات کا مطلب بیان کر سکے
- () عملی زندگی سے مثالوں کی نشاندہی کر سکے
- () موضوعات کی اہمیت بیان کر سکے
- () اپنی شخصیت اور معاشرے پر موضوعات کے مطابق اثرات پیدا کرنے کے طریقے بیان کر سکے
- () مثبت ذہن کے ساتھ کام کر سکے
- () عدل و انصاف سے ادارہ میں، دفتر میں، بہتر ماحول پیدا کر سکے
- () ماحول کو اخلاقی طور پر پاکیزہ بنائے
- () کارکنوں کی بہتر طور پر دل جوئی کر سکے
- () کارکردگی میں اضافہ کر سکے
- () باہمی احترام کی برکات سے استفادہ کر سکے
- () بچوں کے حقوق بیان کر سکے
- () عورتوں کے حقوق کی پاسداری کر سکے
- () بزرگ شہریوں کے حقوق کا خیال رکھ سکے
- () محنت حضرات کے حقوق جان سکے اور ان کے حقوق کی پاسداری کر کے انہیں معاشرے کا ایک مفید فرد بنایا جاسکے

CIT-312 Internet of Things (IoT)

Total Contact Hours

Theory	32	T	P	C
Practical	96	1	3	2

In this course, students will explore the fundamental concepts and capabilities of smart devices and the physical principles behind sensing technologies. They will investigate key components of IoT systems, including enabling technologies, architectures, and standards. A focus will be placed on understanding the IoT protocol stack and addressing constraints and opportunities related to wireless and mobile networks. Students will also examine the fundamentals of Social IoTs and their implications for social interactions and data sharing. The course culminates in a hands-on project where students will apply their IoT knowledge to implement a small-scale IoT solution.

Learning Outcomes: The students will be able to

Learn the concept and capabilities of smart thing/object/device identification, and physical principles of sensing.

- Explore the key components that make up an IoT system i.e., IoT enabling technologies, architectures, and standards.
- Understanding of IoT protocol stack, constraints and opportunities of wireless & mobile networks and fundamentals of Social IoTs.
- Apply IoT knowledge to implement small-scale IoT Project.

COURSE OUTLINE

Contents	Hours
1. Fundamental of Internet of Things (IoT)	08 Hrs
1.1. Definition and characteristics of IoT	
1.2. Requirements, functionalists, applications and structure of IoT	
1.3. IoT enabling technologies	
1.4. M2M Communication	
1.5. IoT Architecture, Core IoT functional stack, Fog, Edge and Cloud in IoT	
1.6. Functional blocks of an IoT ecosystem, sensors, actuators and connecting	
1.7. Sensing components and devices	
2. IoT Protocol Stack	08 Hrs
2.1. Basics of Networking	
2.1.1. Understanding the OSI model and the seven abstraction layers	
2.1.2. Networking and TCP/IP	

2.2. IoT Access Technologies

2.2.1. Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.11ah and Lora WAN

2.2.2. Bluetooth, Wi-Fi and Zigbee

2.3. Network Layer

2.2.1 IP versions, Constrained Nodes and Constrained Networks, 6LoWPAN

2.4. Application Layer Protocols

2.2.2 CoAP, MQTT and AMQP

3. IoT Platforms and Supporting Services

06 Hrs

3.1. Embedded computing logic, Microcontroller, System on Chips, IoT system building blocks

3.2. IoT platform overview

3.3. Overview of IoT supported Hardware platforms such as: Raspberry pi, Arduino board detail

3.4. Real-time systems and embedded software

3.5. IoT supporting services

3.5.1. Computing using a Cloud Platform for IoT/M2M Applications/Services

3.5.2. Everything as a service and Cloud Service Models

4. IoT Security

05 Hrs

4.1. Attack, Defense, and Network Robustness of Internet of Things

4.2. Malware Propagation and Control in IoT

4.3. Privacy Preservation Data Dissemination

4.4. Trust and Trust Models for the IoT

4.5. Authentication in IoT

4.6. Security Protocols for IoT Access Networks

5. Case Studies/Industrial Applications

05 Hrs

5.1. IoT applications in home, infrastructures, buildings, security, Industries

5.2. Home appliances, Model Discussions (Smart Home, Smart Agriculture,

5.3. Smart Retail Supply, Smart Healthcare, Smart Grid, Smart Cities)

5.4. Industry 4.0 concepts

RECOMMENDED BOOKS

- 1 IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, Cisco Press, 2017
- 2 Internet of Things: Architecture, Design Principles and Applications, Rajkamal, McGraw Hill Higher Education
- 3 The Internet of Things (The MIT Press Essential Knowledge series) 2nd Edition by Samuel Greengard

INSTRUCTIONAL OBJECTIVES

1. Fundamental of Internet of Things (IoT)

1.1 Definition and characteristics of IoT

- 1.1.1 Define the terms IoT
- 1.1.2 How Does IoT Work?
- 1.1.3 Discuss the importance of IoT devices

1.2 Requirements, functionalists, applications and structure of IoT

- 1.2.1 Discover the fundamentals of IoT and Cloud Computing
- 1.2.2 Explain the deployment methods of IoT
- 1.2.3 Describe the structure of IoT
- 1.2.4 Analyze potential applications of IoT and the cloud

1.3 IoT enabling technologies

1.4 M2M Communication

- 1.4.1 Introduction to M2M Communication
- 1.4.2 Difference between IoT and M2M
- 1.4.3 Define Software Defined Networking (SDN)
- 1.4.4 Define Network Function Virtualization (NFV) for IoT

1.5 IoT Architecture, Core IoT functional stack, Fog, Edge and Cloud in IoT

- 1.5.1 Define the terms IoT and Cloud Computing
- 1.5.2 Describe the evolution that has led to Edge and Cloud Computing
- 1.5.3 Identify the Components that forms part of IoT Architecture

1.6 Functional blocks of an IoT ecosystem, sensors, actuators and connecting

- 1.6.1 Determine the most appropriate IoT Devices and Sensors based on Case Studies
- 1.6.2 Setup the connections between the Devices and Sensors

1.7 Sensing components and devices

- 1.7.1 Describe the Sensing components and devices
- 1.7.2 Describe the Sensor modules, nodes and systems
- 1.7.3 Use visualization techniques to show data generated from the IoT devices

2. IoT Protocol Stack

2.1 Basics of Networking

- 2.1.1 Understanding the OSI model and the seven abstraction layers
- 2.1.2 Describe the IoT communication model & Protocols
- 2.1.3 Discuss Wireless technologies for the IoT
- 2.1.4 Explain Edge connectivity and protocols
- 2.1.5 Define Wireless sensor networks
- 2.1.6 Networking and TCP/IP
- 2.1.7 Evaluate the appropriate protocol for communication between IoT
- 2.1.8 Analyze the communication protocols for IoT

2.2 IoT Access Technologies

- 2.2.1 Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.11ah and Lora WAN

- 2.2.2 Bluetooth, Wi-Fi and Zigbee

2.3 Network Layer

- 2.3.1 IP versions, Constrained Nodes and Constrained Networks, 6LoWPAN

2.4 Application Layer Protocols

- 2.4.1 CoAP, MQTT and AMQP

3. IoT Platforms and Supporting Services

- 3.1 Embedded computing logic, Microcontroller, System on Chips, IoT system building blocks

- 3.2 IoT platform overview

- 3.3 Overview of IoT supported Hardware platforms such as: Raspberry pi, Arduino board Detail

- 3.3.1 Introduction to Raspberry Pi interfaces (Serial, SPI, I2C)

- 3.3.2 Programming Raspberry Pi with Python, other IoT devices

- 3.4 Real-time systems and embedded software

- 3.5 IoT supporting services

- 3.5.1 Computing using a Cloud Platform for IoT/M2M Applications/Services

- 3.5.2 Implement and secure your own cloud service

- 3.5.3 Everything as a service and Cloud Service Models

4. IoT Security

- 4.1 Attack, Defense, and Network Robustness of Internet of Things

- 4.1.1 Security Issues in the IoT

- 4.1.2 Security Mechanisms Overview

- 4.1.3 Traditional vs Lightweight security

- 4.2 Malware Propagation and Control in IoT

- 4.3 Privacy Preservation Data Dissemination

- 4.3.1 Discuss the Authorization Mechanisms for Secure IoT Services

- 4.3.2 Discuss the Privacy Issues in the IoT

- 4.3.3 Describe the Role of Authorization

- 4.4 Trust and Trust Models for the IoT

- 4.5 Authentication in IoT

- 4.6 Security Protocols for IoT Access Networks

5. Case Studies/Industrial Applications

- 5.1 IoT applications in home, infrastructures, buildings, security, Industries

- 5.2 Home appliances, Model Discussions (Smart Home, Smart Agriculture, Smart Retail Supply, Smart Healthcare, Smart Grid, Smart Cities)

- 5.3 Industry 4.0 concepts

List of Practicals

1. Study the fundamental of IoT software and components.
2. Familiarization with Arduino/Raspberry Pi and perform necessary software configuration
3. To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds.
4. To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection.
5. To interface DHT11 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings.
6. To interface motor using relay with Arduino/Raspberry Pi and write a program to turn ON motor when push button is pressed.
7. To interface OLED with Arduino/Raspberry Pi and write a program to print temperature and humidity readings on it.
8. To interface Bluetooth with Arduino/Raspberry Pi and write a program to send sensor data to smartphone using Bluetooth.
9. To interface Bluetooth with Arduino/Raspberry Pi and write a program to turn LED ON/OFF when '1'/'0' is received from smartphone using Bluetooth
10. Write a program on Arduino/Raspberry Pi to upload temperature and humidity data to ThingSpeak.com cloud.
11. Write a program on Arduino/Raspberry Pi to retrieve temperature and humidity data from ThingSpeak.com cloud.
12. To install MySQL database on Raspberry Pi and perform basic SQL queries
13. Write a program to create UDP server on Arduino/Raspberry Pi and respond with humidity data to UDP client when requested.
14. Write a program to create TCP server on Arduino/Raspberry Pi and respond with humidity data to TCP client

CIT-313 Network & Server Administration

Total Contact Hours

Theory	32	T	P	C
Practical	192	1	6	3

The Course on Network Administration using Microsoft Windows Server 2016 Operating System aims to provide students with practical exercise and knowledge on installing and operational administration of connected system. It covers all major topics of Microsoft Windows Server 2016. This comes with practical component to complement classroom discussion.

Learning Outcomes: The students will be able to

- Describe the basic concepts, architecture, organization and operational principles of Windows Server 2016.
- Install and configure Windows Server 2016
- Manage the resources network users using Windows Server 2016
- Troubleshoot Windows Server 2016 faults and correct them
- Run application programs in Windows Server 2016
- Demonstrate motivation in managing users in a network
- Introduction to DevOps

COURSE OUTLINE

Contents	Hours
1. Introduction to Networks	02 Hrs
1.1 Introduction	
1.2 What is a network	
1.3 Types of Network	
1.4 Major Functions of a Network	
2. Microsoft Windows Client-End	06 Hrs
2.1 Introduction to workgroup & domain	
2.2 Introduction to Client – Server Environment	
2.3 Installation / Deployment of Windows Client end	
2.4 Client End basic settings	
2.5 Basic Components of a Network	
2.6 Client End troubleshooting and maintenance	
3. Installation and Configuration of window server 2016	04 Hrs
3.1 Hardware Requirements of window server 2016	
3.2 Installation of a server and basic Configuration	
3.3 Important roles and features of windows server 2016.	

- | | |
|---|---------------|
| 4. Microsoft Windows Server 2016 Network Services | 10 Hrs |
| 4.1 Major Network Services in windows server 2016 | |
| 4.2 Assigning IP Addressing and Configuring | |
| 4.3 Domain Name systems | |
| 4.4 Installation and Configuring a DNS | |
| 4.5 Installation and Configuring DHCP | |
| 4.6 Connecting to Networks | |
| 4.7 Sharing files, folders and printers | |
| 5. Microsoft Windows Server 2016 Active Directory Infrastructure | 03 Hrs |
| 5.1 Concepts of forest , Tree domain & Subdomain | |
| 5.2 Function and features of an Active Directory | |
| 5.3 Installation of Active Directory | |
| 5.4 Configuration of Active Directory | |
| 5.5 Role of an Active Directory | |
| 5.6 Users, Groups, Computers and organizational units in ADDS | |
| 6. Introduction to DevOps | 07 Hrs |
| 6.1 Introduction to DevOps culture and Process | |
| 6.2 DevOps and Software Development Lifecycle | |
| 6.3 DevOps Stages | |
| 6.4 DevOps culture, collaboration, and team structures | |
| 6.5 Continuous Delivery and Deployment (CD/D) | |

Reference Books.

- MCSA Windows Server 2016 Complete Study guide By William Panek
- Windows Server 2016 Cookbook By Jordan Krause
- Windows Server 2016: Unleashed By Rand Morimoto, Jeffrey Shapiro, Guy Yardeni, Omar Droubi, Michael Noel, Andrew Abbate, Chris Amaris

Instructional Objective

1. Introduction to Networks

- 1.1 Describe Introduction
- 1.2 Describe What is a network
- 1.3 Explain the types of Network
- 1.4 Describe the major Functions of a Network

2. Microsoft Windows Client-End

- 2.1 Give Introduction to workgroup & domain
- 2.2 Introduction to Client – Server Environment
- 2.3 Apply on a Installation / Deployment of Windows Client end
- 2.4 Design a Client End basic settings
- 2.5 Design a Basic Components of a Network
- 2.6 Design a Client End troubleshooting and maintenance

3. Installation and Configuration of window server 2016

- 3.1 Explain and Hardware Requirements of window server 2016
- 3.2 Explain and Installation of a server and basic Configuration
- 3.3 Explain and Important roles and features of windows server 2016.

4. Microsoft Windows Server 2016 Network Services

- 4.1 Explain Major Network Services in windows server 2016
- 4.2 Assigning IP Addressing and Configuring
- 4.3 Domain Name systems
- 4.4 Describe Installation and Configuring a DNS
- 4.5 Installation and Configuring DHCP
- 4.6 Connecting to Networks
- 4.7 Sharing files, folders and printers

5. Microsoft Windows Server 2016 Active Directory Infrastructure

- 5.1 Concepts of forest , Tree domain & Subdomain
- 5.2 Describe Function and features of an Active Directory
- 5.3 Installation of Active Directory
- 5.4 Configuration of Active Directory
- 5.5 Describe Role of an Active Directory
- 5.6 Describe Users, Groups, Computers and organizational units in ADDS

6. Introduction to DevOps

- 6.1 Explain the Introduction to DevOps culture and Process
- 6.2 Benefits of DevOps
- 6.3 Describe DevOps Main Objectives
- 6.4 Describe DevOps and Software Development Lifecycle
- 6.5 DevOps Stages
- 6.6 Describe DevOps culture, collaboration, and team structures
- 6.7 Describe Continuous Delivery and Deployment (CD)

List of Practical's

1. Windows Operating System Installation and Configuration
2. Configuration of IPv4 & IPv6.
3. Study of basic network command and Network configuration commands.
4. Setting up and securing a wireless network (Wi-Fi)
5. Configuring and optimizing wireless access points
6. Connect the computers in Local Area Network.
7. Windows Server 2016 Installation and Configuration
8. DNS Installation, Configuration
9. DHCP installation and configuration
10. NAT Services Configuration and Maintenance
11. Active Directory Installation, Configuration and Maintenance
12. Hard Disk Partition Management in windows operating system
13. Exploring Control Panel for Task manager
14. Installation of Local Printer Implementing basic network security measures (firewalls, antivirus, etc.)
15. Creating and configuring VLANs (Virtual Local Area Networks)
16. Connectivity troubleshooting using PING, IPCONFIG, IFCONFIG
17. Configuring an FTP (File Transfer Protocol) server for file sharing
18. Sharing folders
19. User creation and joining Groups
20. Account Policy and User Rights Management
21. User Profile Implementation
22. Resource Sharing and Security
23. File and Folder Auditing
24. Setting File and Folder Permissions
25. Resource Access and Permission Problems Troubleshooting
26. Backup and Restore the Data and Sharing.
27. Configuration of CD/D.

CIT-314 Mobile Application Development

Total Contact Hours

Theory	64	T	P	C
Practical	192	2	6	4

This Flutter course for mobile app development guides students from basic setup to advanced application deployment, focusing on real-world projects. Participants will learn to create dynamic apps, such as a food ordering platform and a video chatting service, incorporating essential features like user authentication, real-time messaging, and payment gateway integration. The curriculum emphasizes hands-on experience with animations, state management, and backend services like Firebase. Students will also gain proficiency in using local and cloud storage solutions, generating reports, and integrating local financial services. By the end of the course, students will be equipped to build, polish, and deploy feature-rich mobile applications on the popular app stores.

Learning Outcomes: The students will be able to

- Understand the Flutter framework. Build mobile applications for Android and iOS platforms.
- Implement end-to-end app functionalities, including user authentication, real-time messaging, payment gateway integration, and cloud storage, using Firebase and other back-end services.
- Develop skills in creating intuitive and engaging user interfaces, incorporating animations, state management, and adaptive layouts to enhance the overall user experience.
- Execute the process of preparing and deploying mobile applications to the Google Play Store, including generating build files, managing app assets, and navigating the app submission and review process.
- Gain practical experience in developing complex applications such as food ordering and video chatting apps, equipping them with the knowledge to tackle real-world mobile development challenges.

COURSE OUTLINE

Contents	Hours
1. Introduction to smart mobile devices	02 Hrs
1.1 App, GUI, Screens, UI elements	
1.2 Hardware: Camera, GPS, Proximity, Inertial Sensors	
1.3 Wifi and Data Connections	
1.4 Background Worker Threads	
2. Introduction to mobile apps	03 Hrs
2.1 App Screen	
2.2 State, Routes	

2.3	UI Elements	
2.4	User Input: Touch, Pinch, Swipe, Gestures, Keyboard Input	
3.	Flutter Introduction	04 Hrs
3.1.	Dart	
3.2.	Widgets	
3.3.	App Structure	
3.4.	Widget Tree	
3.5.	Flutter Architecture	
4.	Dart Language	05 Hrs
4.1.	Syntax	
4.2.	Types	
4.3.	Conditionals	
4.4.	Control flow	
4.5.	Functions	
5.	Layouts	04 Hrs
5.1.	Container	
5.2.	Row	
5.3.	Column	
5.3.1.	Stacks	
5.3.2.	Grid item	
5.3.3.	Building Basic UI using Widget Library	
6.	Custom Layouts	04 Hrs
6.1.	Customizing Widgets	
7.	Interactivity	04 Hrs
7.1.	Add User Input	
7.2.	Gesture Detection	
7.3.	User Touch	
8.	Navigation	07 Hrs
8.1.	Maintain state based on user interaction	
8.2.	Screen navigation using Navigator + routes	
8.3.	Create multi-screen app using named routes	
8.4.	Passing data between screens	
9.	Animation	05 Hrs
9.1.	Adding animation	
9.2.	Animation library	

9.3. Implicit and explicit navigation

10. Deployment

06 Hrs

- 10.1. Prep for deployment on platform stores (App store and Google Play)
- 10.2. Signing
- 10.3. Packaging
- 10.4. Submission
- 10.5. Best Practices
- 10.6. Plagiarism in code
- 10.7. UI or content
- 10.8. GIT
- 10.9. Documentation
- 10.10. Code security

11. Backend

06 Hrs

- 11.1. Working with local storage (SQLite)
- 11.2. Firebase
- 11.3. Authentication using Firebase
- 11.4. Firestore DB cloud storage

12. Backend

05 Hrs

- 12.1. API integration
- 12.2. http request handling in app

13. Backend

02 Hrs

- 13.1. JSON
- 13.2. JSON parsing

14. Introduction to Java for Android

07 Hrs

- 14.1. Syntax
- 14.2. Control
- 14.3. Types
- 14.4. Conditionals
- 14.5. Functions
- 14.6. Classes
- 14.7. Comparison with JavaScript
- 14.8. Comparison with C++
- 14.9. Packages and reuse
- 14.10. Examples of two open source small java apps.

RECOMMENDED BOOK

1. "Flutter for Beginners" by Alessandro Biessek
2. "Flutter in Action" by Eric Windmill
3. "Beginning Flutter: A Hands On Guide to App Development" by Marco L. Napoli
- "Pragmatic Flutter" by Priyanka Tyagi

INSTRUCTIONAL OBJECTIVES

1. Introduction to smart mobile devices
 - 1.1. App, GUI, Screens, UI Elements
 - 1.2. Hardware: Camera, GPS, Proximity, Inertial Sensors
 - 1.3. Wifi and Data Connections
 - 1.4. Background Worker Threads
2. Introduction to mobile apps
 - 2.1. App Screen
 - 2.2. State, Routes
 - 2.3. UI Elements
 - 2.4. User input: Touch, Pinch, Swipe, Gestures, Keyboard Input
3. Flutter introduction
 - 3.1. Dart
 - 3.2. Widgets
 - 3.3. App structure
 - 3.4. Widget tree
 - 3.5. Flutter architecture
4. Dart language
 - 4.1. Syntax
 - 4.2. Types
 - 4.3. Conditionals
 - 4.4. Control flow
 - 4.5. Functions
5. Layouts
 - 5.1. Container
 - 5.2. Row, column
 - 5.3. Stacks
 - 5.4. Grid item
 - 5.5. Building basic UI using widget library
6. Custom layouts
 - 6.1. Customizing widgets
7. Interactivity
 - 7.1. Add user input
 - 7.2. Gesture detection
 - 7.3. User touch
8. Navigation
 - 8.1. Maintain state based on user interaction
 - 8.2. Screen navigation using navigator + routes
 - 8.3. Create multi-screen app using named routes

- 8.4. Passing data between screens
- 9. Animation
 - 9.1. Adding animation
 - 9.2. Animation library
 - 9.3. Implicit and explicit navigation
- 10. Deployment
 - 10.1. Prep for deployment on platform stores (app store and google play)
 - 10.2. Signing
 - 10.3. Packaging
 - 10.4. Submission
 - 10.5. Best practices, plagiarism in code, UI or content, GIT, documentation, code security
- 11. Backend
 - 11.1. Working with local storage (SQLite)
 - 11.2. Firebase
 - 11.3. Authentication using firebase
 - 11.4. Firestore DB cloud storage
- 12. Backend
 - 12.1. API integration
 - 12.2. http request handling in app
- 13. Backend
 - 13.1. JSON
 - 13.2. JSON parsing
- 14. Introduction to java for android
 - 14.1. Syntax
 - 14.2. Control
 - 14.3. Types
 - 14.4. Conditionals
 - 14.5. Functions
 - 14.6. Classes
 - 14.7. Comparison with JavaScript
 - 14.8. Comparison with C++
 - 14.9. Packages and reuse
 - 14.10. Examples of two open source small java apps.

LIST OF PRACTICALS

1. Practical

Setup Flutter development environment

2. Practical

Install GIT:

Download and install git from git-scm.com.

Configure git with your name and email.,

`git config --global user.name "Your Name"`

`git config --global user.email "youremail@edu.com.pk"`

Initialize a git Repository ""LAB3""

Create a new directory LAB3 and navigate to it

Initialize the git repository using git init

Design a screen in figma and save to this folder

Check the Status and Log using git status and git log

3. Practical

"Working with git:

Create a new repository (repo) for this lab on github.com

Connect Local Repository to Remote

Add the remote repository to your local git configuration:

`git remote add origin https://github.com/yourusername/my-first-git-repo.git`

Push your local commits to the remote repository:

`git push -u origin master`

Clone the repo using git clone on github.com

4. Practical

Collaboration in git:

Create a New Branch using git checkout -b branchname

Design more figma screens and put it in local repo folder

Make changes to a file (e.g., README.md):

Add and commit the changes:

Push the Branch to Remote:

Create a Pull Request (PR) on github:

Merge the PR

5. Practical

Continuous Integration/Continuous Deployment (CI/CD) with GitHub Actions

Make a simple web project for your personal website in a new directory

Initialize a git Repository in the folder

Commit your work, attach to remote and push

Set Up GitHub Actions for CI/CD

create a .github/workflows directory and a new workflow file deploy.yml
name: Deploy to GitHub Pages

```

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Deploy to GitHub Pages
        uses: peaceiris/actions-gh-pages@v3
        with:
          github_token: ${ secrets.GITHUB_TOKEN }
          publish_dir: ./

```

Commit and Push the Workflow:

Enable GitHub Pages:

Check the Workflow, Go to the "Actions" tab on GitHub to see the CI workflow running. Verify that the deployment succeeds. Visit the URL <https://yourusername.github.io/your-project> to see your live site.

6. Practical

Number guessing game lab series: Make a single page flutter app:

Main page/screen design using figma for a mobile screen, create repo, attach remote, push your work at each step.

Now make screen in flutter according to design. git it.

Add a splash screen, git it.

Add a second option screen to choose range of number, then change the screen to main

7. Practical

Number guessing game lab series:

Create a number guessing game in Flutter with the following features:

Design the UI in Figma. Design the Game Screen, Main screen, history screen, Export Design Assets,

Implement the game in Flutter. Set Up Your Flutter Project,

Add Dependencies, Add sqflite and path dependencies to your pubspec.yaml:

dependencies:

```

flutter:
  sdk: flutter
  sqflite: ^2.0.0
  path: ^1.8.0

```

Create the Main Game Screen, Add SQLite for Local Storage

Create a Guess model:

```
class Guess {  
  final int id;  
  final int number;  
  final String result;  
  Guess({this.id, this.number, this.result});  
  Map<String, dynamic> toMap() {  
    return {  
      'id': id,  
      'number': number,  
      'result': result,  
    };  
  }  
}
```

Create a SQLite helper class:

```
import 'package:sqflite/sqflite.dart';  
import 'package:path/path.dart';
```

```
class DatabaseHelper {  
  static final DatabaseHelper _instance = DatabaseHelper._internal();  
  factory DatabaseHelper() => _instance;  
  DatabaseHelper._internal();  
  Database _database;  
  Future<Database> get database async {  
    if (_database != null) return _database;  
    _database = await _initDatabase();  
    return _database;  
  }  
  Future<Database> _initDatabase() async {  
    String path = join(await getDatabasesPath(), 'guesses.db');  
    return openDatabase(  
      path,  
      onCreate: (db, version) {  
        return db.execute(  
          """CREATE TABLE guesses(id INTEGER PRIMARY KEY AUTOINCREMENT, number  
INTEGER, result TEXT)""",  
        );  
      },  
      version: 1,  
    );  
  }  
  Future<void> insertGuess(Guess guess) async {
```

```

final db = await database;
await db.insert('guesses', guess.toMap(), conflictAlgorithm: ConflictAlgorithm.replace);
}

```

```

Future<List<Guess>> getGuesses() async {
  final db = await database;
  final List<Map<String, dynamic>> maps = await db.query('guesses');
  return List.generate(maps.length, (i) {
    return Guess(
      id: maps[i]['id'],
      number: maps[i]['number'],
      result: maps[i]['result'],
    );
  });
}

```

Implement Game Logic:

Display Guess Score History

Version control the project using Git.

8. Practical

Number guessing game Lab series: Modify and make improvements, GIT it.

9. Practical

Number guessing game Lab series

Make layout better adjust size, fonts and color etc

git it

Pull previous version and run to confirm if improvement is undo-ed

Pull latest version again and run to confirm if improved UI is back again

10. Practical

Number word guessing Hangman game Lab series

Make Figma design for Hangman game, git it.

Convert design to Flutter screens app, then git it.

Improve design, playability, git it

11. Practical

Number word guessing Hangman game Lab series:

Complete and improve

12. Practical

Expense Tracker App Lab Series: Create a comprehensive expense tracker app in Flutter, progressing from basic in-memory storage to advanced features including local database storage, multi-screen navigation, reporting, PDF export, charts, cloud storage, and authentication using Firebase, and location API integration.

By following these steps, you'll build a feature-rich expense tracker app in Flutter. Starting with basic in-memory storage, you'll advance to using SQLite for local storage, and then expand to a multi-screen app with reporting features. Finally, you'll add advanced functionalities like charting, Firebase

integration for cloud storage and authentication, and location API integration.

Step 1: In-Memory Expense Tracking

Setup and Basic UI

- Create a new Flutter project.
- Design a simple UI for adding and viewing expenses (fields: date, amount, description).
- Store expenses in a list in memory.
- Display the list of expenses on the main screen.

Step 2: Local Database Storage with SQLite

Integrate SQLite:

- Add `sqlite` and `path` dependencies.
- Create a SQLite helper class to handle database operations.
- Modify the app to save expenses to a local SQLite database instead of in-memory.
- Implement functions to add, retrieve, update, and delete expenses from the database.

Step 3: Multi-Screen Navigation

Multi-Screen App:

- Create separate screens for adding expenses, viewing expense history, and viewing reports.
- Implement navigation between screens using Flutter's Navigator.
- Pass data between screens as needed.

Step 4: Reporting and PDF Export

Expense Reports:

- Create daily, weekly, and monthly expense reports.
- Use the `flutter\_pdf` package to generate PDFs for each report.
- Add an export button to save the PDF to the device.

Step 5: Expense Charts

Integrate Charts:

- Add `fl\_chart` or a similar charting library.
- Create bar and pie charts to visualize daily, weekly, and monthly expenses.
- Display charts on the report screen.

Step 6: Cloud Storage and Authentication with Firebase

Firebase Integration:

- Add Firebase to your Flutter project.
- Implement Firebase Authentication for user login and registration.
- Use Firebase Firestore to store and sync expenses in the cloud.
- Ensure that each user's data is securely stored and retrieved from their respective Firebase account.

Step 7: Location API Integration

Location API:

- Integrate the `location` package to capture the user's location when adding an expense.
- Store the location data with each expense entry.
- Display the location data in the expense details.

13. Practical

"Employee Attendance app Lab Series: Create a comprehensive attendance tracking app in Flutter, progressing from basic user time-in/time-out functionality to advanced features including user management, reporting, admin interface, shift management, alerts, and report exports.

Step 1: Basic Time-In/Time-Out Functionality

Setup and Basic UI:

- Create a new Flutter project.
- Design a simple UI for users to log time-in and time-out using their username and a 4-digit pincode.
- Store time-in and time-out data in memory.
- Display a confirmation message upon successful logging of time.

Step 2: User Management

Add New User Functionality:

- Implement a screen for adding new users with fields for username and 4-digit pincode.
- Store user data in a list in memory.
- Ensure new users can log time-in and time-out after being added.

Step 3: Basic Reports

Generate User Reports:

- Create screens to display user reports including total Hrs worked and absentees.
- Implement basic calculations to determine Hrs worked based on time-in and time-out logs.
- Display a list of users who have not logged any Hrs for a given day as absentees.

Step 4: Admin Interface

Admin Interface and Reports:

- Implement an admin login screen with a username and pincode.
- Create an admin dashboard to view all user logs and generate detailed reports.
- Include options for filtering and sorting logs by date, user, and Hrs worked.

Step 5: Shift Management and Alerts

Add Shifts and Alerts:

- Implement functionality to define shifts and assign them to users.
- Update reports to include shift information.
- Implement alert notifications for missed shifts or discrepancies in logging.

Step 6: Export and Final Features

Report Exports and Final Features:

- Add functionality to export reports in PDF or CSV format.
- Use a package like `flutter\_pdf` or `csv` to handle report generation.
- Polish the app with final UI improvements and ensure all features work seamlessly.

14. Practical

"Patient Doctor appointment App Lab Series: Create a comprehensive patient-doctor appointment app in Flutter, starting with basic appointment booking for a single doctor and progressing to advanced features like multiple doctors, customized slot calendars, appointment management, and advanced reporting. By completing these steps, you'll build a comprehensive patient-doctor appointment app in Flutter. You'll start with basic appointment booking, progress to implementing multiple doctors and customized slot calendars, and finally add advanced features like appointment management and

reporting. This lab ensures a thorough understanding of building a feature-rich appointment management application from start to finish.

Step 1: Basic Appointment Booking

Setup and Basic Appointment UI:

- Create a new Flutter project.
- Design a simple UI for patients to book appointments with a single doctor using fixed time slots.
- Store appointment data in memory.
- Display a confirmation message upon successful booking.

Step 2: Doctor Screen with Auto-Update

Implement Doctor Screen:

- Create a doctor screen to display all booked appointments.
- Use a state management solution (e.g., Provider) to auto-update the doctor's screen when new appointments are booked.
- Display appointment details including patient name and time slot.

Step 3: User Authentication

Login and Multiple Doctors:

- Implement a login screen for patients and doctors using username and password.
- Extend the app to support multiple doctors.
- Store user data in memory and authenticate users on login.

Step 4: Customized Slot Calendar

Customized Slot Calendar:

- Implement a calendar view for doctors to customize their available time slots.
- Allow doctors to define their availability for different days.
- Ensure the booking screen reflects the customized slots.

Step 5: Appointment Management

Cancel and Retract Appointment:

- Implement functionality for patients to cancel or retract their appointments.
- Update the doctor's screen to reflect these changes.
- Ensure canceled slots become available for other patients to book.

Step 6: Advanced Reporting

Generate Advanced Reports:

- Create screens to display advanced reports such as total appointments per day, week, and month.
- Include filters for specific doctors and time periods.
- Implement functionality to export these reports in PDF or CSV format using packages like ``flutter_pdf`` or ``csv``.

15. Practical

"Game Lab Series: Chrome browser game like hurdle jump and duck game with a dinosaur character, with score storage, leaderboard, user auth. Create an engaging and visually appealing dinosaur hurdle jump and duck game in Flutter, focusing on animations and graphics development along with game mechanics, score storage, leaderboards, and user authentication. By completing these steps, you'll build

a dynamic and visually appealing dinosaur hurdle jump and duck game in Flutter. You'll focus heavily on animations and graphics development, while also implementing essential game mechanics, scoring, user authentication, and leaderboards. This lab ensures a thorough and engaging learning experience in building an animated game from start to finish.

Step 1: Basic Game Setup and Graphics

Setup and Basic Graphics:

- Create a new Flutter project.
- Design the game background and ground.
- Create and animate a simple dinosaur character using sprite sheets or custom graphics.
- Implement basic dinosaur movements (jumping and ducking) using animations.

Step 2: Hurdle Obstacles and Movement

Obstacle Design and Movement:

- Design various hurdle obstacles (e.g., cacti, birds).
- Implement the movement of hurdles across the screen.
- Use animations to create a seamless and dynamic obstacle appearance.
- Ensure hurdles spawn at random intervals and speeds to increase game difficulty.

Step 3: Game Logic and Collision Detection

Implement Game Logic and Collision Detection:

- Develop the game logic for jumping and ducking actions.
- Implement collision detection between the dinosaur and hurdles using bounding boxes or collision detection libraries.
- Create animations for collision effects (e.g., flashing screen, explosion).

Step 4: Scoring System and Animations

Scoring and Feedback Animations:

- Implement a scoring system that increases the score as the dinosaur successfully avoids hurdles.
- Display the current score on the screen with animated transitions.
- Add visual feedback for scoring milestones (e.g., glowing effects, particle systems).

Step 5: User Authentication and Leaderboards

User Authentication and Leaderboards:

- Integrate Firebase Authentication for user login and registration.
- Store high scores in Firebase Firestore, linked to authenticated users.
- Create a leaderboard screen that displays top scores with animated transitions and user avatars.
- Update the leaderboard in real-time as users achieve new high scores.

Step 6: Game Polishing and Final Features

Polish and Finalize Game Features:

- Add background music and sound effects for jumping, ducking, collisions, and scoring.
- Improve the overall game UI/UX with smooth animations and transitions.
- Implement a game over screen with animated score display and restart options.

- Ensure the game works smoothly on both Android and iOS devices, optimizing animations and graphics for performance.

16. Practical

"Building a Food Banda Online Food Ordering App Lab Series: Create a comprehensive online food ordering app for Pakistan's local financial services like JazzCash and EasyPaisa, starting from a simple customer interface and progressing to advanced features like multiple login roles, item management, reporting, and payment gateway integration. By completing these steps, you'll build a comprehensive and fully functional online food ordering app tailored for Pakistan's local financial services. Starting with a simple customer interface, you'll progress to adding essential features like cart management, multiple login roles, item and restaurant management, reporting, and integration with JazzCash and EasyPaisa payment gateways. This lab ensures a thorough understanding of building a feature-rich food ordering application from start to finish. Git each lab in this series in the same repo.

Step 1: Basic Customer Interface with Hardcoded Items

Setup and Basic UI:

- Create a new Flutter project.
- Design a simple UI for a single store with a list of hardcoded food items.
- Display item details (name, description, price) and an "Add to Cart" button for each item.

Step 2: Add to Cart and Review Cart

Implement Add to Cart and Cart Review:

- Implement functionality to add items to the cart.
- Create a cart screen where customers can review their selected items, update quantities, and see the total cost.
- Ensure items in the cart are persistent across app sessions using local storage.

Step 3: Checkout and Billing

Implement Checkout and Billing:

- Design a checkout screen that displays the cart summary and allows customers to enter billing details.
- Implement a simple order confirmation process that simulates order placement and generates an order summary.

Step 4: Multiple Login Roles

Implement Login and Roles:

- Integrate Firebase Authentication for user login and registration.
- Define roles for customers, restaurant staff, and admins.
- Implement role-based navigation to restrict access to specific screens and functionalities based on user roles.

Step 5: Item and Restaurant Management

Item and Restaurant Management:

- Implement functionality for restaurant staff to add new items, edit existing items, and manage their store details.
- Create an admin interface to add new restaurants and oversee all items and stores.
- Ensure changes made by restaurant staff and admins are reflected in the customer interface.

Step 6: Reporting for Admins and Merchants

Generate Reports:

- Create reporting screens for admins and restaurant staff.
- Admin reports should include total orders, revenue, and customer statistics.
- Merchant reports should include order details, item performance, and daily earnings.

Step 7: Integration with Local Payment Gateways

Payment Gateway Integration:

- Integrate with JazzCash and EasyPaisa payment gateways.
- Implement secure payment processing on the checkout screen.
- Ensure successful payment transactions update the order status and generate a payment receipt.

Step 8: Final Checkout Process

Finalize and Polish Checkout and User Experience:

- Refine the checkout process to handle various payment methods, including cash on delivery.
- Add order tracking and status updates for customers post-checkout.
- Ensure the entire app is tested for smooth user experience and optimized for performance.
- Make it more easy to use and look good on the eyes.

Step 9: Deploy on store

Deploy app on Google play store:

- Sign the package and upload on the store.

17. Practical

"Video chatting app Lab series: Create a comprehensive video chatting app in Flutter, starting with user registration and progressing through text messaging, voice calling, video calling, and finally deploying the app to the Google Play Store. By completing these steps, you'll build a fully functional video chatting app in Flutter. Starting from user registration, you'll advance through text messaging, voice and video calling, user presence, push notifications, user settings, call history, and finally deploy the app to the Google Play Store. This comprehensive lab series ensures a thorough understanding of building and deploying a video chatting application from start to finish.

Step 1: Project Setup and User Interface

Setup and Basic UI:

- Create a new Flutter project.
- Design a simple login and registration screen.
- Implement Firebase Authentication for user registration and login.

Step 2: User Registration and Authentication

User Registration and Authentication:

- Set up Firebase Authentication to handle user registration and login.
- Create a user profile screen that displays user information.

Step 3: User Presence and Status

Implement Online/Offline Status:

- Use Firebase Realtime Database or Firestore to track user online/offline status.
- Display the online status of users on the user profile screen.

Step 4: Contact List

Create Contact List:

- Implement a screen to display a list of registered users.
- Allow users to add other users to their contact list.

Step 5: Text Messaging

Implement Text Messaging:

- Use Firebase Firestore for real-time messaging.
- Create a chat screen for sending and receiving text messages.
- Implement basic text message functionalities (send, receive, display messages).

Step 6: User Presence in Messaging

Show User Presence in Chat:

- Display online/offline status in chat screens.
- Indicate when a user is typing.

Step 7: Push Notifications for Messages

Implement Push Notifications:

- Set up Firebase Cloud Messaging (FCM) for push notifications.
- Implement push notifications for new messages.

Step 8: Voice Call Functionality

Implement Voice Call:

- Integrate a VoIP service like Agora or Twilio.
- Create a voice call screen.
- Implement basic voice calling functionalities (initiate, receive, end calls).

Step 9: Voice Call Notifications

Voice Call Notifications:

- Implement push notifications for incoming voice calls.
- Ensure users receive notifications when the app is in the background.

Step 10: Video Call Functionality

Implement Video Call:

- Integrate video call functionalities using a service like Agora or Twilio.
- Create a video call screen.
- Implement basic video calling functionalities (initiate, receive, end calls).

Step 11: In-Call User Interface

In-Call UI Enhancements:

- Add features like mute/unmute, switch camera, and end call to the in-call UI.
- Ensure a smooth transition between voice and video calls.

Step 12: User Settings

Implement User Settings:

- Create a settings screen for users to update their profile information and preferences.
- Allow users to toggle notifications and call settings.

Step 13: Call History

Implement Call History:

- Track and display a history of voice and video calls.
- Store call logs in Firebase Firestore.

Step 14: User Block and Report

User Block and Report Feature:

- Implement functionality for users to block or report other users.
- Ensure blocked users cannot contact the reporting user.

Step 15: App Testing and Debugging

Test and Debug the App:

- Thoroughly test all app functionalities on different devices.
- Debug and fix any issues to ensure a smooth user experience.

Step 16: Deployment to Google Play Store

Deploy to Google Play Store:

- Prepare app assets (icons, screenshots).
- Set up a Google Play Developer account.
- Follow the steps to build and sign the APK.
- Upload the app to the Google Play Store and complete the necessary store listing information.
- Submit the app for review and monitor the release process."

18. Practical

Expense app Lab Series: Remake expenses app in java

19. Practical

Expense app Lab Series: Integrate Flutter app with part of Java based expense app.

CIT- 322 Entrepreneurship & Freelancing

Total Contact Hours

Theory	32	T	P	C
Practical	96	1	3	2

This curriculum provides a comprehensive guide to IT entrepreneurship, beginning with an introduction to the field and the characteristics of successful IT entrepreneurs. It explores the startup ecosystem and market gap recognition, followed by the process of validating and refining business ideas. It delves into market research, customer discovery, competition analysis, and various business models in IT, along with revenue streams and monetization strategies. The curriculum also covers drafting business plans, financial projections, pitching, bootstrapping, and understanding funding options such as angel investors and venture capital. Additionally, it includes a detailed section on freelancing, covering various platforms, profile building, proposal writing, pricing strategies, time management, and balancing multiple projects.

Learning Outcomes: The students will be able to

- Understand basic concept of IT Entrepreneurship
- Understand market research and business models of IT
- Understand and search startup ideas and funding options
- Create account on different online markets
- Write proposal for clients in online markets

COURSE OUTLINE

Contents	Hours
1. Introduction to IT Entrepreneurship	03 Hrs
1.1. Overview of entrepreneurship in the IT sector	
1.2. Key characteristics of successful IT entrepreneurs	
1.3. Understanding the startup ecosystem	
1.4. Recognizing market gaps in IT	
1.5. Validating and refining business ideas	
2. Market Research and Business Model	09 Hrs
2.1. Conducting effective market research in IT	
2.2. Customer discovery and validation processes	
2.3. Analyzing competition and identifying unique value propositions	
2.4. Exploring different business models in IT such as SaaS, PaaS, IaaS, etc	
2.5. Revenue streams and monetization strategies	

- | | |
|--|---------------|
| 3. Technology Startup & Funding Options | 04 Hrs |
| 3.1. Explain all critical aspects to draft a comprehensive business plan | |
| 3.2. Describe Financial projections and budgeting for IT startups | |
| 3.3. Explain pitching and presenting the business plan | |
| 3.4. Explain Bootstrapping and ways to self-fund startups. | |
| 3.5. What is Angel investors and venture capital? | |
| 4. Freelancing | 06 Hrs |
| 4.1. What is Freelancing? | |
| 4.2. Explain different fields in freelancing | |
| 4.3. List down different Freelancing platforms | |
| 4.3.1. Describe Personal attitude in Freelancing | |
| 4.3.2. How to build Profile on Freelancer.com | |
| 4.3.2.1. How to build People per Hour Profile | |
| 4.3.2.2. How to write a proposal and cover letter | |
| 4.3.3. How to build Profile on Fiverr. | |
| 4.3.3.1. How to Create Gigs. | |
| 4.4. Introduction to Upwork | |
| 4.4.1. How to create Upwork profile. | |
| 4.4.2. What are Connects and Membership offers | |
| 4.4.3. Differentiate between Hourly and Fixed projects | |
| 4.4.4. How to find the right job and understanding the clients demand. | |
| 5. Pricing Strategies and Time Management | 04 Hrs |
| 1.1 How to determining fair market value | |
| 1.2 What are the negotiation techniques for freelancers? | |
| 1.3 How to handle project scope and accommodate project changes | |
| 1.4 How to create a freelance schedule. | |
| 1.5 List down tools for time tracking and project management | |
| 1.6 How to Balance multiple projects using Art board | |
| 6. Proposal Writing | 06 Hrs |
| 6.1. Explain ways of studying client's requirements | |
| 6.2. How to read client Mindset. | |
| 6.3. Describe techniques to write a winning proposal | |
| 6.4. Explain how to accommodate client requirement. | |
| 6.5. How to develop a Methodology | |
| 6.6. Explain way to outshine competitors | |
| 6.7. Explain and provide Formal Proposal templates. | |

RECOMMENDED BOOKS

1. User Guide Fiverr
2. User Guide Upwork
3. User Guide Freelancer

INSTRUCTIONAL OBJECTIVES

1. Introduction to IT Entrepreneurship

- 1.1 Explain the fundamental concepts of entrepreneurship in the field of IT Sector and identify major trends and opportunities.
- 1.2 Explain essential characteristics and behaviors that contribute to success in IT entrepreneurship.
- 1.3 Explain startup ecosystem, including key players, resources and process that supports IT growth.
- 1.4 Explain market gaps in IT.
- 1.5 Explain validation methods like customer feedback, prototyping and testing.

2. Market Research and Business Model

- 2.1 Explain the methods of conducting effective market research in IT
- 2.2 Describe customer discovery to identify target market
- 2.3 How to analyze and identify unique value propositions
- 2.4 Explain current business model such as
 - 2.4.1 Software as a service
 - 2.4.2 Platform as a service
 - 2.4.3 Infrastructure as a service
 - 2.4.4 On-Premises Software
 - 2.4.5 Open Source Software
 - 2.4.6 Freemium
 - 2.4.7 Ad-Supported
 - 2.4.8 E-Commerce Websites
 - 2.4.9 Subscription Box Service
- 2.5 Revenue streams and monetization strategies

3. Technology Startup & Funding Options

- 3.1 Components of a comprehensive business plan
- 3.2 Financial projections and budgeting for IT startups
- 3.3 Pitching and presenting the business plan
- 3.4 Bootstrapping and self-funding
- 3.5 Angel investors and venture capital
- 3.6 Crowd funding and alternative financing methods

4. Freelancing

- 3.1 Introduction to Freelancing
- 3.2 Different fields in freelancing
- 3.3 Introduction to Freelancing websites
 - 3.5.4 Personal attitude in Freelancing
 - 3.5.5 Profile building on Freelancer.com

- 3.5.6 Profile building on People Per Hour
 - 3.5.7 Writing a proposal and cover letter
 - 3.6 Introduction to Upwork
 - 3.6.1 Upwork profile creation
 - 3.6.2 Connects and Membership offers
 - 3.6.3 Hourly and Fixed projects
 - 3.6.4 Knowing your customers to find the right jobs
- 5. **Pricing Strategies and Time Management**
 - 5.1 Determining fair market value
 - 5.2 Negotiation techniques for freelancers
 - 5.3 Handling scope creep and project changes
 - 5.4 Creating a freelance schedule
 - 5.5 Tools for time tracking and project management
 - 5.6 Balancing multiple projects Art board
- 6. **Proposal Writing**
 - 6.1 Study the requirements
 - 6.2 Understand the client Mindset
 - 6.3 Writing a winning proposal
 - 6.4 Understanding of the client's requirements.
 - 6.5 Develop a Methodology
 - 6.6 Outshine Competitors
 - 6.7 Formal Proposal

LIST OF PRACTICAL

1. Creating Account on Fiverr
2. Building Profile on Fiverr
3. Web Scraping to add data to fiver profile
4. Creating a Gig on Fiverr
5. Fiverr Packages (Basic, Standard and Premium)
6. Order completion and cancellation procedure
7. A to Z management of LinkedIn
8. Creating proposal for Buyer on standard format
9. Creating Account on Up work
10. Connects and memberships on upwork
11. Hourly and fixed projects
12. Creating Account of Guru.com
13. Building profile on guru.com
14. how to Bid on projects
15. How to use ChatGPT by using perfect prompts
16. Freelancing on Social Media platforms (Facebook, Instagram)
17. Creating account on YouTube
18. Uploading video content on YouTube
19. Awareness about Copyrights & Ethics on Social Media

CIT- 343 Microprocessor Fundamentals

Total Contact Hours

Theory	64	T	P	C
Practical	96	2	3	3

In this course, students will delve into the architecture and organization of the Intel 8086 microprocessor, gaining a comprehensive understanding of its structure and functionality. They will learn to classify and utilize the microprocessor's instruction set, mastering assembly language programming through practical application. The course will cover various addressing modes used in instructions, enabling students to effectively interpret and implement them. Additionally, students will explore and classify different types of interrupts, understanding their role and management within the 8086 microprocessor system.

Learning Outcomes: The students will be able to

- Describe the architecture & organization of Intel 8086 Microprocessor.
 - Understand and classify the instruction set of 8086 microprocessor and distinguish the Use of different instructions and apply it in assembly language programming.
 - Relate the addressing modes used in the instructions.
 - Understand and classify Interrupts
-

COURSE CONTENTS

1. Fundamental Concepts 06 Hrs

- 1.1. Historical background of microprocessor
- 1.2. Basic Terms used in Microprocessor
- 1.3. Organization of a Microprocessor-Based System
- 1.4. Bus based microprocessor

2. Intel 8086 Microprocessor Architecture & Hardware Specifications 12 Hrs

- 2.1. Block Diagram of Intel 8086 Microprocessor
- 2.2. Programming model of 8086
- 2.3. Power and Clocking Requirements of 8088/86
- 2.4. Pin Configuration of 8086
- 2.5. Clock generator (8284A)
- 2.6. Bus Multiplexing and De-Multiplexing
- 2.7. Minimum Mode and Maximum Mode

3. Data Addressing Modes 10 Hrs

- 3.1. Immediate Addressing Mode
- 3.2. Register Addressing Mode

- 3.3. Direct Addressing Mode
- 3.4. Base-Plus-Index Addressing
- 3.5. Register Relative Addressing
- 3.6. Base Relative-Plus-Index Addressing
- 3.7. Program Memory addressing Mode
- 3.8. Stack Memory-Addressing Mode

4. 8086 Instruction Set 12 Hrs

- 4.1. Data Movement/Transfer Instructions
- 4.2. Arithmetic Group of Instructions
- 4.3. Logical Group of Instructions
- 4.4. Program Execution Transfer instructions
- 4.5. String Manipulation Instructions
- 4.6. CALL, RETURN

5. Interrupts 12 Hrs

- 5.1. Purpose of Interrupts
- 5.2. Interrupt Types
- 5.3. Basic interrupt processing
- 5.4. Interrupt Instructions
- 5.5. Interrupt Service Routine
- 5.6. Interrupt Vector Table
- 5.7. Programmable interrupt controller (8259A)

6. I/O Interfacing 12 Hrs

- 6.1. Overview of I/O Interfacing
- 6.2. Ways of Communication – Microprocessor with the Outside World
- 6.3. 8279 – Programmable Keyboard
- 6.4. Intel 8255A – Programmable Peripheral Interface
- 6.5. Intel 8253 – Programmable Interval Timer
- 6.6. 8257 – DMA Controller

RECOMMENDED BOOKS

- 1. Barry B. Brey, “The Intel Microprocessors - Architecture, Programming, and
- 2. Interfacing”, Eight Edition, Pearson Prentice Hall
- 3. Ramesh Gaonkar, “Microprocessor Architecture, Programming and
- 4. Kip R. Irvine , Assembly Language for x86 Processors, Prentice Hall , 6/E, 2011

INSTRUCTIONAL OBJECTIVES

1. Fundamental Concepts

- 1.1. Historical background of microprocessor
 - 1.1.1. Describe the Evolution of Different Microprocessors from first (4-bit) to fifth Generation (64-bit)
- 1.2. Basic Terms used in Microprocessor
 - 1.2.1. Bit, Word, Bus, System Bus, Memory Word, Bus width, Microcontroller, Microcomputer
 - 1.2.2. Difference between Microprocessor & Microcontroller
 - 1.2.3. Microcomputer Block Diagram
- 1.3. Organization of a Microprocessor-Based System
 - 1.3.1. Define Arithmetic and Logic Unit (ALU), Register Array, Control unit
 - 1.3.2. Define Memory and its types
 - 1.3.3. Define I/O Devices and its types
- 1.4. Bus based microprocessor
 - 1.4.1. Address, Data and Control Buses

2. Intel 8086 Microprocessor Architecture & Hardware Specifications

- 2.1. Block Diagram of Intel 8086 Microprocessor
 - 2.1.1. Explain each part of block diagram
 - 2.1.2. Describe Advantages of Architecture of 8086
 - 2.1.3. Describe Dis-advantages of Architecture of 8086:
- 2.2. Programming model of 8086
 - 2.2.1. Function and working of each part of programming model
 - 2.2.2. Real mode memory addressing
 - 2.2.3. Protected mode memory addressing
 - 2.2.4. Memory Paging
- 2.3. Power and Clocking Requirements of 8088/86
- 2.4. Pin Configuration of 8086
 - 2.4.1. Describe the function of each pin
- 2.5. Clock generator (8284A)
 - 2.5.1. Pin Diagram of 8284A
 - 2.5.2. Function of Clock Generator 8284A
- 2.6. Bus Multiplexing and De-Multiplexing
 - 2.6.1. Need of Bus multiplexing & De-Multiplexing
 - 2.6.2. Fully buffered 8086 diagram
- 2.7. Minimum Mode and Maximum Mode of operation

3. Data Addressing Modes

- 3.1. Immediate Addressing Mode
- 3.2. Register Addressing Mode
- 3.3. Direct Addressing Mode
- 3.4. Base-Plus-Index Addressing
- 3.5. Register Relative Addressing
- 3.6. Base Relative-Plus-Index Addressing
- 3.7. Program Memory addressing Modes
- 3.8. Stack Memory-Addressing Modes

4. 8086 Instruction Set

- 4.1. Data Movement/Transfer Instructions
 - 4.1.1. MOV, XCHG, PUSH, POP, PUSHF, POPF, PUSHA, POPA, IN, OUT
- 4.2. Arithmetic Group of Instructions
 - 4.2.1. ADD, ADC, SUB, SBB, INC, DEC, MUL, IMUL, DIV, IDIV, DAS, AAM, AAA
- 4.3. Logical Group of Instructions
 - 4.3.1. AND, OR, NOT, XOR, SHR, SHL, ROR, ROL, TEST
- 4.4. Program Execution Transfer instructions
 - 4.4.1. CALL, RET, JMP, LOOP
- 4.5. String Manipulation Instructions
 - 4.5.1. REP/ REPE/ REPZ/ REPNE/ REPNZ, MOVS/ MOVSB/ MOVSN, CMPS/ CMPSB/ CMPSW, SCAS/ SCASB/ SCASW, LODS/ LODSB/ LODSW, STOS/ STOSB/ STOSW
- 4.6. CALL, RETURN

5. Interrupts

- 5.1. Purpose of Interrupts
- 5.2. Interrupt Types
 - 5.2.1. Difference between Software and Hardware Interrupts
 - 5.2.2. Maskable & Non-Maskable Interrupts
- 5.3. Basic interrupt processing
- 5.4. Interrupt Instructions
 - 5.4.1. BOUND, INTO, INT, INT-3, IRET
- 5.5. Interrupt Service Routine
- 5.6. Interrupt Vector Table
- 5.7. Programmable interrupt controller (8259A)
 - 5.7.1. Features of 8259 Programmable Interrupt Controller
 - 5.7.2. Block and Pin diagram of 8259A

6. I/O Interfacing

- 6.1. Overview of I/O Interfacing
- 6.2. Define Memory Interfacing, IO Interfacing
- 6.3. Ways of Communication – Microprocessor with the Outside World
 - 6.3.1. Describe Serial Communication and Parallel Communication interface
- 6.4. Describe 8279 – Programmable Keyboard interface with 8086
- 6.5. Describe Intel 8255A – Programmable Peripheral Interface
- 6.6. Describe Intel 8253 – Programmable Interval Timer
- 6.7. 8257 – DMA Controller
- 6.8. Describe the Sequence of Operations Performed by a DMA

List of Practical

1. Introduction to Flowcharting
2. Algorithm Design Using Flowcharts
3. Assembly Language Program Structure
4. Introduction to MASM (Editor, Assembler, Linker, Debugger, Debugger Function).
5. Introduction How to Create A Source Code
6. Understanding the Registers of 8086 Microprocessor.
7. Use of 8086 Instructions.
8. Demonstrating MOV Instruction.
9. Demonstrating XCHG Instruction.
10. Demonstrating PUSH Instruction.
11. Demonstrating POP Instruction.
12. Demonstrating PUSHF Instruction.
13. Demonstrating POPF Instruction.
14. Demonstrating ADD Instruction.
15. Demonstrating SUB Instruction.
16. Demonstrating MUL Instruction.
17. Demonstrating DIV Instruction.
18. Demonstrating INC Instruction.
19. Demonstrating DEC Instruction.
20. Demonstrating NOT Instruction.
21. Demonstrating AND Instruction.
22. Demonstrating OR Instruction.
23. Demonstrating XOR Instruction.
24. Understanding the Addressing Modes of 8086 Microprocessor.
25. Implementing the Immediate Addressing Modes of 8086 Microprocessor.
26. Implementing the Register Addressing Modes of 8086 Microprocessor.
27. Understanding the Memory Addressing Mode.
28. Understanding Memory Direct Addressing Mode.
29. Using Memory Direct Addressing Mode to Transfer Data (5h, 10h, 15h, 20h, 25h) to Memory.
30. Using Memory Direct Addressing Mode to Transfer Data (5h, 10h, 15h, 20h, 25h) from Memory to Register.
31. Using Register Indirect Addressing Mode to Transfer Data (5h, 10h, 15h) Memory to Register.
32. Write a Program For Ascending Order.
33. Write a Program For Descending Order.
34. Using Trainer and Instructions of Assembly Language Control Traffic Light.

CIT-372 UI/UX Design

Total Contact Hours

Theory	32	T	P	C
Practical	96	1	3	2

This course introduces students to the exciting world of UI/UX Design, focusing on hands-on learning with Figma, a popular design tool. Through practical exercises and projects, students will gain the foundational skills to create user-friendly and visually appealing interfaces for websites and applications.

Learning Outcomes: The students will be able to

- Understand the basic principles of UI/UX design.
- Apply design thinking to create user-friendly interfaces.
- Utilize Figma for prototyping and creating basic user interfaces.

COURSE OUTLINE

Contents	Hours
1. Introduction to UI/UX Design	04 Hrs
1.1. What is UI/UX Design? (User Interface & User Experience)	
1.2. Why is UI/UX Design important? (Impact on users & businesses)	
1.3. Importance of User-Centered Design	
2. The Design Process	04 Hrs
2.1. Empathize	
2.2. Define	
2.3. Ideate	
2.4. Prototype	
2.5. Test	
3. User Interface (UI) Design	10 Hrs
3.1. Design Principles (Balance, Contrast, Hierarchy, etc.)	
3.2. UI Elements (Buttons, Forms, Icons, etc.)	
3.3. Typography and Color Theory	
4. User Experience (UX) Design	10 Hrs
4.1. User Research (Understanding User Needs)	
4.2. User Personas (Creating fictional representations of target users)	
4.3. User Flows & Sitemaps (Visualizing user journeys & content structure)	
4.4. Basic Usability Testing (Identifying user pain points)	

5. Wireframing & Prototyping

04 Hrs

- 5.1. Wireframing - Creating low-fidelity representations of layouts.
- 5.2. Prototyping - Simulating app or website functionality.

Reference books

- 2024 - 2025 Newbies Guide to UI/UX Design Using Figma - Anthony E Sanchez

INSTRUCTIONAL OBJECTIVES

1. Introduction to UI/UX Design

- 1.1. Explain What is UI/UX Design? (User Interface & User Experience)
- 1.2. Explain Why is UI/UX Design important? (Impact on users & businesses)
- 1.3. Explain Importance of User Centered Design

2. The Design Process

- 2.1. Explain how to Empathize
- 2.2. Explain how to Define
- 2.3. Explain how to Ideate
- 2.4. Explain how to Prototype
- 2.5. Explain how to Test

3. User Interface (UI) Design

- 3.1. Explain Design Principles (Balance, Contrast, Hierarchy, etc.)
- 3.2. Explain UI Elements (Buttons, Forms, Icons, etc.)
- 3.3. Explain Typography and Color Theory

4. User Experience (UX) Design

- 4.1. Explain User Research (Understanding User Needs)
- 4.2. Explain User Personas (Creating fictional representations of target users)
- 4.3. Explain User Flows & Sitemaps (Visualizing user journeys & content structure)
- 4.4. Explain Basic Usability Testing (Identifying user pain points)

5. Wireframing & Prototyping

- 5.1. Explain Wireframing - Creating low-fidelity representations of layouts.
- 5.2. Explain Prototyping - Simulating app or website functionality.

List of Practical's

1. Create your profile

- a. Signup with Figma
- b. Create your Profile Using AutoLayout.
- c. Instructions: <https://www.figma.com/community/file/798962767266141877>

2. Make a heart

- a. Introduce how to make different shapes using pen tool, layout grid and properties.
- b. Ask the students to make a heart shape, fill with a color
- c. Instructions: <https://www.figma.com/community/file/828040090909268885>

3. Create a Chat Window

- a. Create a chat window using autolayout
- b. Instructions: <https://www.figma.com/community/file/829875886550676425>

4. Create Loading Animations

- a. Ask students create animations that they would like to show to the user while something is being loaded of some processing is being done.
- b. Instructions: <https://www.figma.com/community/file/826886430526815045>

5. A grid system for presentations

- a. This lab will allow the students to learn about layout and grid systems
- b. Instructions: <https://www.figma.com/community/file/831257037874341872>

6. Railway Ticketing APP - Wireframes

- a. Create Wireframes for the Railway Ticketing APP that has following screens:
 - i. **Home Screen:** Displays prominent sections for searching trains, booking tickets, checking PNR status, and user profile.
 - ii. **Search Screen:** Allows users to input departure and arrival stations, date, number of passengers, and train preferences.
 - iii. **Train Search Results:** Displays a list of available trains with details like departure/arrival times, availability, fare, etc.
 - iv. **Ticket Booking:** Collects passenger details, seat preferences, payment information, and generates a booking confirmation.
 - v. **PNR Status:** Allows users to check the status of their booked tickets using PNR number.
 - vi. **User Profile:** Displays user information, booking history, and account settings.
- b. Resource: <https://www.figma.com/templates/wireframe-kits/>

7. Railway Ticketing APP - Actual Screens

- a. Now Create Actual Screens for the Railway Ticketing APP
- b. Use existing icons and company logo taken from the internet as you like
- c. Instructions:

Establish a Design System:

- **Color Palette:** Choose a color palette that reflects the railway brand identity.
- **Typography:** Select legible fonts for headings, body text, and buttons.
- **Spacing:** Determine consistent spacing units for elements.
- **Iconography:** Design or select icons for key functionalities.

Create Layouts:

- Design the layout for each screen, considering content hierarchy and visual balance.
- Utilize Figma's Auto Layout feature to create responsive layouts that adapt to different screen sizes.

Implement Responsiveness:

- Design the app to function effectively on various screen sizes (small, medium, large).
- Prioritize essential content on smaller screens and provide details on larger screens.
- Use breakpoints to define layout changes at specific screen widths.

8. Railway Ticketing App - Dark Mode

- a. Make a copy of the Railway Ticketing APP and make the following changes
 - **Dark Mode:** Change the color scheme including the design of any icons to dark mode.
- b. Ensure that the APP displays perfectly in both Normal Mode and Dark Mode.

9. Railway Ticketing APP - Icons

- a. Now design custom icons for the Railway Ticketing APP
- b. The icon should be developed for the following:

i. Home	vii. Notifications
ii. Search	viii. Payment
iii. Train	ix. PNR status
iv. Ticket	x. Location
v. Calendar	xi. Clock
vi. User profile	xii. Information
- c. Instructions:

Choose an Icon Style: Decide on a consistent icon style (outlined, filled, flat, etc.) that aligns with the app's overall design aesthetic.

Create Icon Sketches: Begin by sketching out your icon ideas on paper to explore different concepts.

Digitalize Icons: Use Figma to create digital versions of your icon sketches.

Refine and Iterate: Continuously refine your icons based on feedback and usability considerations.

Ensure Consistency: Maintain consistency in icon size, style, and color palette

throughout the set.

- d. Ensure that you create logos for both normal and dark modes.

10. Railway Ticketing Company - Logo

- a. Design the logo for the Railway Ticketing Company
- b. The logo needs to be added to the Railway Ticketing APP
- c. Instructions:

Understand the Brand: Research the railway company's values, target audience, and brand personality.

Brainstorm Concepts: Generate ideas for logo concepts based on the company's identity and the railway theme. Consider elements such as:

- i. Trains, locomotives, or tracks
- ii. Speed, travel, or journey
- iii. Reliability, safety, or comfort
- iv. Modernity or heritage

Sketching: Create rough sketches of potential logo designs.

Digitalization: Use Figma to create digital versions of selected sketches.

Typography: Choose a suitable font that complements the logo concept and reflects the company's personality.

Color Palette: Select colors that evoke the desired emotions and align with the brand identity.

Logo Variations: Create different versions of the logo for various applications (horizontal, vertical, black and white).

Testing and Refinement: Seek feedback on the logo designs and make necessary adjustments.

11. Online Darzi Web APP - Wireframes

- a. Create Wireframes for the Railway Ticketing APP that has following screens:
 - i. **Homepage:** This is the first impression of your website. It should clearly communicate the services offered, include a strong call to action (e.g., "Book Now"), and showcase customer testimonials or before-and-after images.
 - ii. **About Us:** This page should introduce the business, its mission, values, and team. Consider including information about the experience and expertise of the tailors.
 - iii. **Services:** Clearly outline the services offered, such as stitching, alterations, embroidery, etc. Include detailed descriptions, pricing information, and images of finished products.
 - iv. **Portfolio:** Showcase a gallery of the best work to demonstrate the quality of craftsmanship. Categorize the portfolio by service type or clothing item.
 - v. **Pricing:** Clearly display the pricing structure for different services.

- Consider adding a pricing calculator for custom orders.
- vi. **Contact Us:** Provide multiple contact options (phone, email, address, social media) and a contact form for inquiries.
 - vii. **Booking:** A user-friendly booking form that collects customer details, order specifications, and preferred delivery or pickup options.
 - viii. **Order Tracking:** A page where customers can track the status of their orders.
 - ix. **FAQs:** Address common customer questions and concerns.
- b. Resource: <https://www.figma.com/templates/wireframe-kits/>

12. Online Darzi Web APP - Actual Screens

- a. Now Create Actual Screens for the Online Darzi Web APP
- b. Use existing icons and company logo taken from the internet as you like
- c. Instructions:

Establish a Design System:

- **Color Palette:** Choose a color palette that reflects the railway brand identity.
- **Typography:** Select legible fonts for headings, body text, and buttons.
- **Spacing:** Determine consistent spacing units for elements.
- **Iconography:** Design or select icons for key functionalities.

Create Layouts:

- Design the layout for each screen, considering content hierarchy and visual balance.
- Utilize Figma's Auto Layout feature to create responsive layouts that adapt to different screen sizes.

Implement Responsiveness:

- Design the app to function effectively on various screen sizes (small, medium, large).
- Prioritize essential content on smaller screens and provide details on larger screens.
- Use breakpoints to define layout changes at specific screen widths.

13. Online Darzi Web APP - Responsive

- a. Make a copy of the Railway Ticketing APP and make the following changes
- b. Ensure that the APP adapts seamlessly to different screen sizes
- c. Test your APP with the following screen sizes:
 - i. Desktop: 1280 x 720, 1920 x 1080
 - ii. Tablet: 601 x 962, 1280 x 800
 - iii. Mobile: 360 x 640, 414 x 896
- d. Make modifications to adjust the layout for all sizes.

14. Online Darzi Web APP - Dark Mode

- a. Make a copy of the Online Darzi Web APP and make the following changes
 - **Dark Mode:** Change the color scheme including the design of any icons to dark mode.
- b. Ensure that the APP displays perfectly in both Normal Mode and Dark Mode.

15. Online Darzi Web APP - Icons

- a. Now design custom icons for the Online Darzi Web APP
- b. The icon should be developed for the following:
 - i. Home
 - ii. Services
 - iii. Portfolio
 - iv. About Us
 - v. Contact Us
 - vi. Booking
 - vii. Cart (if applicable)
 - viii. Account (if applicable)
 - ix. Search
 - x. Menu (for mobile view)
- c. Instructions:

Choose an Icon Style: Decide on a consistent icon style (outlined, filled, flat, etc.) that aligns with the app's overall design aesthetic.

Create Icon Sketches: Begin by sketching out your icon ideas on paper to explore different concepts.

Digitalize Icons: Use Figma to create digital versions of your icon sketches.

Refine and Iterate: Continuously refine your icons based on feedback and usability considerations.

Ensure Consistency: Maintain consistency in icon size, style, and color palette throughout the set.

- d. Ensure that you create logos for both normal and dark modes.

16. Online Darzi Company - Logo

- a. Design the logo for the Online Darzi Company
- b. The logo needs to be added to the Online Darzi Web APP
- c. Instructions:

Understand the Brand: Research the railway company's values, target audience, and brand personality.

Brainstorm Concepts: Generate ideas for logo concepts based on the company's identity and the railway theme. Consider elements such as:

- i. Trains, locomotives, or tracks
- ii. Speed, travel, or journey
- iii. Reliability, safety, or comfort
- iv. Modernity or heritage

Sketching: Create rough sketches of potential logo designs.

Digitalization: Use Figma to create digital versions of selected sketches.

Typography: Choose a suitable font that complements the logo concept and reflects the company's personality.

Color Palette: Select colors that evoke the desired emotions and align with the brand identity.

Logo Variations: Create different versions of the logo for various applications (horizontal, vertical, black and white).

Testing and Refinement: Seek feedback on the logo designs and make necessary adjustments.

CIT-381 Software Quality Assurance

Total Contact Hours

Theory	0	T	P	C
Practical	96	0	3	1

The course is structured to assist students in learning key topics in Software Quality Assurance, such as software testing objectives, processes, criteria, strategies, and methodologies. The course also covers subjects like how to create test cases and test data, how to execute testing operations, how to manage software problems and defects, software test automation using real world examples, and how to perform software security and performance testing.

Learning Outcomes: The students will be able to

- Learn the concept and capabilities of software testing and software quality Assurance principles
- Prepare test case and test suites for completely testing all aspects of a System Under Test (SUT)
- Compile findings of a quality assurance cycle
- Carry out inspections and carry out testing in a production environment

COURSE CONTENTS

1. Introduction to quality assurance
2. Software development life cycles (SDLC)
3. Quality assurance phases
4. Python programming basics
5. Automation testing – basics
6. Automation testing – advanced
7. Test script, database and SQL – basics
8. Performance test – basics
9. Security test – basics

LIST OF PRACTICALS

1. Test Scenario Preparation
2. Test Case Preparation with Selenium
3. Test Data Preparation with Selenium
4. Decision Table Preparation with Selenium
5. Behavior-driven development (BDD) Examples Practice
6. Test Script Preparation (SQL)
7. Browsing Web via Automation using Playwright
8. Browsing Web via Automation using Scrapy
9. Mobile application testing with Appium
10. Data Validation via Automation
11. Web Scraping using Selenium
12. Accessing Multiple URLs using Selenium
13. Create Keyword Driven Automation Framework
14. Create Automation Test Result in Excel
15. Performance Test using JMeter
16. Performance Testing using Calabash
17. Web security pen-testing

CIT-392 Project

Total Contact Hours

Theory	0	T	P	C
Practical	192	0	6	2

This proctored hands-on course guides participants through the end-to-end process of IOT/Web/ Mobile app development. Starting from idea discussion and research, students will engage in iterative design and implementation, incorporating continuous feedback and version control. Participants will develop a fully functional application, refine UI/UX, and ensure thorough testing and bug fixing. The course culminates with a polished project presentation, ready for industry review. This will be conducted in teams of 3 to 4 persons.

Learning Outcomes: The students will be able to

At the end of the course, the students are expected to be able to exhibit

- Develop and refine app ideas through research and discussion.
- Design user interfaces with iterative feedback and improvement.
- Architect and implement backend systems, including database design and third-party API integration.
- Utilize version control (GIT) for project management and collaboration.
- Create and present working prototypes and progressively enhance functionality.
- Conduct thorough testing and bug fixing to ensure a stable, user-friendly application.
- Improve the app's UI/UX based on testing and feedback.
- Prepare and deliver a final project presentation ready for industry review.
- IOT Based Project
- Project Proposal with Write Up
- Evaluation of Project Proposal for Feasibility
- Consultation & Present

Project Activities

1. Proposal

- Define the project's purpose and objectives.
- Outline the scope and deliverables
- Identify users and their expectations.

2. Planning

- Develop a detailed project plan with tasks and deadlines.
- Allocate resources (e.g., tools, team members).

3. Research

- Conduct market research.
- Gather requirements from user.
- Analyze existing solutions (web/mobile/IOT)

- 4. UI and Design**
 - Create wireframes for the UI.
 - Design the overall system architecture.
- 5. Prototype**
 - Build an initial prototype to demonstrate main features.
- 6. Development**
 - Implement the project in phases.
 - Integrate components and features step by step.
 - Conduct weekly team meetings with teacher to track progress.
- 7. Testing**
 - Perform automated and manual testing.
 - Identify and resolve bugs or issues.
- 8. Review**
 - Gather feedback from teachers, students and users
 - Make revisions based on feedback and test results.
- 9. Final Evaluation**
 - Assess the final product against the project goals.
 - Compile a final project report detailing outcomes
- 10. Presentation**
 - Prepare a comprehensive presentation of the project.
 - Highlight key features, achievements, and challenges.
 - Deliver the presentation to evaluators.

MINIMUM QUALIFICATION OF TEACHERS/ INSTRUCTORS

- M.Sc.IT/ MCS (Master in Computer Science) with 1-Year relevant experience in teaching/ industry
- BS (Information Technology/Computer Science/Software Engineering/Artificial Intelligence/Data Science/ Computer Engineering/Cyber Security or Equivalent) with 2-Years' relevant industrial/teaching experience
- DAE CIT/ ICT / Software Tech., with 6-Years' relevant Industrial / Teaching experience.

For Subjects pertaining to Humanities / Related Studies /Management

- MA degree in English/ Islamiat / Pst. with 2 years of teaching experience
- M.Sc./B.Sc. degree in IT / Computer Sciences or BE in Computer Engineering with 2-years of teaching experience
- M.Sc. degree in Mathematics with 2-years of teaching experience
- M.Sc. degree in Physics / Chemistry with 2-years of teaching experience
- MBA degree with 2-years of teaching experience

POTENTIAL EMPLOYABILITY OF THE PASS-OUTS

The students holding DAE Computer information technology can seek employment in firms/companies / industry dealing in any of the following Areas / Sectors:-

- Industrial Automation and Controls
- Hardware Support
- Devops and Network Administrator
- Freelancing & Entrepreneurship

Job Occupations:

- Software Engineer
- Python developer
- Web Developer
- App developer
- Software Quality Engineer
- IOT Engineer
- UI/UX Designer
- Graphic Designer
- Database Administrator / Engineer

DETAIL OF COMMITTEE

S. No.	<u>Name & Designation</u>	<u>Qualification</u>	<u>Experience</u>
1	Dr. Junaid Arshad , Associate Professor Department of Computer Science, UET Lahore	Phd Computer Science	23 Year
2	Dr. Talha Waheed , Assistant Professor Department of Computer Science, UET Lahore	Phd Computer Science	28 Year
3	Muhammad Amir Iqbal , Consultant (Raheem Solution) Pvt.Ltd. Lahore	MS Computer Science	25 Year
4	Dr. Khawaja Muhammad Umar Suleman Director (Automatrix) Lahore	Phd Computer Science	24 Year
5	Tayyba Amin Sr. Instructor GCT for PGA, Lahore	B.Sc. Elect. Engg. M.Sc Telecommunication	19 Year
6	Engr. Tahir Javed Instructor GCT Taxila	M.Sc. Computer Engineering	13 Years
7	Hafiz Usman Dilshad Jr. Instructor GCT Bahawalpur	Mphil Computer Science	10 Years