Government of Pakistan

National Vocational and Technical Training Commission

Prime Minister's Hunarmand Pakistan Program

"Skill for All"

Course Contents / Lesson Plan

Course Title: Software Development (PYTHON / JAVA / SQL / C++)

Duration: 3 Months

Course Details / Description & Preliminaries

Trainer Name	
Course Title	**Software Development (PYTHON / JAVA / SQL / C++)**
Objectives and Expectations	**Employable skills for Bachelor of Computer Science (BCS) / B.Sc (Computer Science) through an intensive course on software development (PYTHON /JAVA/SQL)** This course is designed with a special purpose in mind - to address the issue of unemployment in the youth. Our team of dedicated professionals, who have rich market/work experience, will deliver hands-on practical training to help achieve this objective. Unlike traditional courses that focus on developing only theoretical knowledge, this course aims to equip trainees with the skills necessary to perform effectively in a commercial environment, either as independent professionals or as part of a team. In addition to technical skills, we also focus on developing soft skills, such as interpersonal and communication skills, and personal grooming, as well as entrepreneurial skills, such as marketing and freelancing. Our course also emphasizes the importance of work ethics, which not only fosters better citizenship but also improves the image of the Pakistani workforce. **Main Expectations**: In short, the course under reference is designed to be delivered in a robust, hands-on manner by professional instructors to equip trainees with practical skills for wage-earning or self-employment. This approach goes beyond traditional training practices and emphasizes a market-centric perspective as the driving force. The instructors must have extensive experience and knowledge to identify the training needs for potential market roles and assess the strengths and weaknesses of each trainee to prepare them for success in the job market. At the conclusion of the course, trainees should feel confident in their abilities to apply their skills and earn a living.

Key Features of Training& Special Modules	**i.** We have included specially designed practical tasks for trainees to perform and enhance their skills. We encourage you to preserve records of all tasks performed, including the name, trade, and session, so that we can monitor your progress effectively. **ii.** In addition, we have included a special module on Job Search & Entrepreneurial Skills. This module will teach you job search techniques, including those used in the local and international job markets (such as the Gulf countries). You will also gain awareness about the visa process and immigration laws of preferred labor destinations. Furthermore, we will encourage you to explore self-employment opportunities and teach you the requirements to start your own business. **iii.** We also believe in inculcating a sense of civic duty, roles, and responsibilities among our trainees to make them responsible citizens of the country. As part of this, we have included a module on Workplace Ethics, outlining the importance of good behavior and positive attitude in the workplace. To make this module engaging and interesting, we have incorporated PPT slides and short video documentaries to convey its significance. By putting our hearts and souls into the course and emphasizing these non-technical components, we aim to positively transform the image of Pakistani workforce in both local and international job markets. In order to maintain interest and motivation of the trainees throughout the course, modern techniques such as: • Motivational Lecture • Success Stories • Case Studies These techniques would be employed as an additional training tool wherever possible (these are explained in the subsequent section on Training Methodology). Lastly, evaluation of the competencies acquired by the trainees will be done objectively at various stages of the training and proper record of the same will be maintained. Suffice to say that for such evaluations, practical tasks would be designed by the training providers to gauge the problem solving abilities of the trainees.

Training Tools/ Methodology	**(i) Motivational Lecture** The proposed methodology for the training under reference employs motivation as a tool. Hence besides the purely technical content, a trainer is required to include elements of motivation in his/her lecture_to inspire the trainees to utilize the training opportunity to the full and strive towards professional excellence. Motivational lectures may also include general topics such as the importance of moral values and civic role & responsibilities as a Pakistani. A Motivational Lecture should be delivered with enough zeal to produce a deep impact on the trainees. It may comprise of the following: • Clear Purpose to convey message to trainees effectively. • Personal Story to quote as an example to follow. • Trainees Fit so that the situation is actionable by trainees and not represent a just idealism. • Ending Points to persuade the trainees on changing themselves. A good Motivational Lecture should help drive creativity, curiosity and spark the desire needed for trainees to want to learn more. Impact of a successful motivational strategy is amongst others commonly visible in increased class participation ratios. It increases the trainees' willingness to be engaged on the practical tasks for longer time without boredom and loss of interest because they can clearly see in their mind's eye where their hard work would take them in short (1-3 years); medium (3 -10 years) and long term (more than 10 years). As this tool is expected that the training providers would make arrangements for regular well planned Motivational Lecture*(For further detail please see Page No: 4)*s as part of a coordinated strategy interspersed throughout the training period as suggested in the weekly lesson plans in this document. Course related motivational lectures online link is available in **Annexure-II.** **(ii) Success Stories** Another effective way of motivating the trainees is by means of Success Stories. Its inclusion in the weekly lesson plan at regular intervals has been recommended till the end of the training. A success story may be disseminated orally, through a presentation or by means of a video/documentary of someone that has risen to fortune, acclaim, or brilliant achievement. A success story shows how a person

achieved his goal through hard work, dedication and devotion. An inspiring success story contains compelling and significant facts articulated clearly and easily comprehendible words. Moreover, it is helpful if it is assumed that the reader/listener knows nothing of what is being revealed. Optimum impact is created when the story is revealed in the form of:-

- Directly in person (At least 2-3 cases must be arranged by the training institute)
- Through an audio/ videotaped message (2-3 high quality videos must be arranged by the training institute)

It is expected that the training provider would collect relevant high quality success stories for inclusion in the training as suggested in the weekly lesson plan given in this document.

Suggestive structure and sequence of a sample success story and its various shapes can be seen at annexure III.

(iii) Case Studies

Where a situation allows, case studies can also be presented to the trainees to widen their understanding of the real life specific problem/situation and to explore the solutions.

In simple terms, the case study method of teaching uses a real life case example/a typical case to demonstrate a phenomenon in action and explain theoretical as well as practical aspects of the knowledge related to the same. It is an effective way to help the trainees comprehend in depth both the theoretical and practical aspects of the complex phenomenon in depth with ease. Case teaching can also stimulate the trainees to participate in discussions and thereby boost their confidence. It also makes class room atmosphere interesting thus maintaining the trainee interest in training till the end of the course.

Depending on suitability to the trade, the weekly lesson plan in this document may suggest case studies to be presented to the trainees. The trainer may adopt a power point presentation or video format for such case studies whichever is deemed suitable but it's important that only those cases are selected that are relevant and of a learning value.

The Trainees should be required and supervised to carefully analyze the cases.

For the purpose they must be encouraged to inquire and collect specific information / data, actively participate in the discussions and intended solutions of the problem / situation.

Case studies can be implemented in the following ways:-

	i. A good quality trade specific documentary (At least 2-3 documentaries must be arranged by the training institute) ii. Health & Safety case studies (2 cases regarding safety and industrial accidents must be arranged by the training institute) iii. Field visits (At least one visit to a trade specific major industry/ site must be arranged by the training institute)
Learning Outcome of the Course	After completion of this course, the trainees must be able to: ● Knowledge and skills in compute, networking, storage, and database AWS services as well as AWS deployment and management services ● Knowledge and skills in deploying, managing, and operating workloads on AWS as well as implementing security controls and compliance requirements ● Knowledge and skills in using the AWS Management Console and the AWS Command Line Interface (CLI)Understanding of the AWS Well-Architected Framework, AWS networking, security services, and the AWS global infrastructure ● Ability to identify which AWS services meet a given technical requirement and to define technical requirements for an AWS-based application
Entry level of trainees	Since intake level is Bachelor of Computer Science (BCS) / B.Sc (Computer Science) so expectations of the trainees are: • Understanding of programming concepts and experience in a programming language such as C#, Java, JavaScript, Python, or Ruby. • Understanding of fundamental programming concepts such as data types, control structures, and functions. • Familiarity with fundamental cloud computing concepts and services, such as storage, networking, compute, and security. • Understanding of AWS core services, including compute, storage, database, and networking.
Course Execution	Total Duration of Course: 3 Months (12 Weeks)

Plan	Class Hours: 4 Hours per day (05 Days/Week)
	Theory: 20% Practical: 80%
	Weekly Hours: 20 Hours Per week, Monthly Hours: 80
	Total Contact Hours: 240 Hours
Companies Offering Jobs in the respective trade	Besides overseas employment, the following Pakistani companies/firms/Organizations are also offering jobs as well, with details as under:- ● Freelancing (Fiverr, Upwork, Freelancers) ● Government Institutes ● Arbisoft Pvt. Ltd ● Crossover ● Techlogix ● Netsol Technologies ● KeepTruckin ● Cloud Technology Solutions ● Nayatel Pvt. Ltd ● 10Pearls ● Teradata ● Systems Limited ● CodeJunkie Pvt. Ltd ● TeraMera Pvt. Ltd ● Software Houses etc
Job Opportunities	Completing the course can provide good job opportunities not only in Pakistan as well as outside the country, as the course is associated with high-demand skills in the industry. 1. AZ-204: Developing Solutions for Microsoft Azure: After

	completing this certification, an individual can work as a Cloud Developer, Azure Solutions Architect, Cloud Solutions Engineer, or Cloud Consultant, etc. Many top IT companies in Pakistan such as Systems Limited, Cognizant Technology Solutions, and Techlogix, etc. offer job opportunities to certified Azure professionals. 2. Certified Associate in Python Programming (Exam PCAP-31-03): Python is widely used in the IT industry, and after completing this certification, an individual can work as a Python Developer, Data Scientist, or Machine Learning Engineer, etc. Companies such as Abacus Consulting, KeepTruckin, and S&P Global Market Intelligence, etc. offer job opportunities to certified Python professionals in Pakistan. 3. SAACO3 (AWS Certified Solutions Architect - Associate): AWS is one of the most popular cloud platforms globally, and after completing this certification, an individual can work as an AWS Solutions Architect, Cloud Engineer, or Cloud Consultant, etc. Many top IT companies in Pakistan such as Systems Limited, Ciklum Pakistan, and Techlogix, etc. offer job opportunities to certified AWS professionals.
No of Students	25
Learning Place	Classroom / Lab

WEEKLY SCHEDULE OF TRAINING

Scheduled Week	Module Title	Day	Hours	Learning Units	Home Assignments
Certified Associate in Python Programming (Exam PCAP-31-03)					
Week 1	Introduction to software development with python	**Day 1**	**Hour 1**	• Motivational Lecture	
			Hour 2	• Course Introduction	
			Hour 3	• Success stories	
			Hour 4	• Job market	
	Introduction to software development with python	**Day 2**	**Hour 1**	• Course Application in Industry	
			Hour 2	• Institute/work ethics	
			Hour 3	• Introduction to Software Develoment	
			Hour 4	• Objectives / Roles	
Week1	Section 1: Modules and Packages (12%) **Objectives covered by the block (6 exam items)**	**Day 3**	**Hour 1-2**	**PCAP-31-03 1.1 – Import and use modules and packages** • import variants: import, from import, import as, import * • advanced qualifying for nested modules • the dir() function • the sys.path variable	**Task 1** *Details may be seen at Annexure-I*

			Hour 3-4	**PCAP-31-03 1.2 – Perform evaluations using the math module** • functions: ceil(), floor(), trunc(), • factorial(), hypot(), sqrt()	
	Section 2: Exception Handling **Objectives covered by the block (5 exam items)**	**Day 4**	**Hour 1**	**PCAP-31-03 1.3 – Generate random values using the random module** • functions: random(), seed(), choice(), sample()	
			Hour 2	**PCAP-31-03 1.4 – Discover host platform properties using the platform module** • functions: platform(), machine(), processor(), system(), version(), • python_implementation(), python_version_tuple()	
			Hour 3-4	**PCAP-31-03 1.5 – Create and use user-defined modules and packages** • idea and rationale; • the __pycache__ directory • the __name__ variable • public and private variables • the __init__.py file • searching for/through modules/packages	

				• **• nested packages vs. directory trees**	
		Day 5	**Hour 1-2**	**PCAP-31-03 2.1 – Handle errors using Python-defined exceptions** • except, except:-except, except:-else:, except (e1, e2) • the hierarchy of exceptions • raise, raise ex • assert • event classes • except E as e • the arg property	
			Hour 3-4	**PCAP-31-02 2.2 – Extend the Python exceptions hierarchy with self-defined exceptions** • self-defined exceptions • defining and using self-defined exceptions	
Week 2	**Section 3: Strings**	**Day 1**	**Hour 1**	• **Feedback of the previous Task** Objectives covered by the block (8 exam items)	• **Task 2** • **Task 3** *Details may be seen at Annexure-I*

			Hour 2-3	**PCAP-31-03 3.1 – Understand machine representation of characters** • encoding standards: ASCII, UNICODE, UTF-8, code points, escape sequences	
			Hour 4	**PCAP-31-03 3.2 – Operate on strings** • functions: ord(), chr() • indexing, slicing, immutability	
		Day 2	**Hour 1-2**	• iterating through strings, concatenating, multiplying, comparing • (against strings and numbers) • operators: in, not in	
			Hour 3-4	**PCAP-31-03 3.3 – Employ built-in string methods** • methods: .isxxx(), .join(), .split(), .sort(), sorted(), .index(), .find(), .rfind()	
		Day 3	**Hour 1-2**	**PCAP-31-03 4.1 – Understand the Object-Oriented approach** • ideas and notions: class, object, property, method, encapsulation, • inheritance, superclass, subclass, identifying class components	

			Hour 3-4	**PCEP-31-03 4.2 – Employ class and object properties** • instance vs. class variables: declarations and initializations • the __dict__ property (objects vs. classes)	
	Section 4: Object-Oriented Programming Objectives covered by the block (12 exam items)	**Day 4**	**Hour 1**	• private components (instances vs. classes) • name mangling	
			Hour 2	***MCQ Type having 30 questions***	
			Hour 3-4	Feedback of the previous Task Motivational Lecture **PCAP-31-03 4.3 – Equip a class with methods** • declaring and using methods • the self-parameter	
		Day 5	**Hour 1-2**	**PCAP-31-03 4.4 – Discover the class structure** • introspection and the hasattr() function (objects vs classes)	
			Hour 3-4	• properties: __name__, __module__ , __bases__	

Week 3	**Section5: Miscellaneous**	**Day 1**	**Hour 1-4**	**PCAP-31-03 4.5 – Build a class hierarchy using inheritance** • single and multiple inheritance • the isinstance() function • overriding • operators:	• **Task 5** • **Task 6** • **Task 7** • **Task 8** *Details may be seen at Annexure-I*
		Day 2	**Hour 1-2**	• not is, is • polymorphism • overriding the __str__() method diamonds	
			Hour 3	**PCAP-31-03 4.6 – Construct and initialize objects** • declaring and invoking constructors	
			Hour 4	***MCQ Type having 30 questions with maximum 4 options per question (Time Allowed: 40 Minutes)***	
		Day 3	**Hour1-2**	**Feedback of the previous Task** Scope: List Comprehensions, Lambdas, Closures, and I/O Operations	
			Hour 3-4	• Objectives covered by the block (9 exam items	
		Day 4	**Hour 2**	**PCAP-31-03 5.1 – Build complex lists using list comprehension** • list	

				comprehensions: the if operator, nested comprehensions	
			Hour 3	**PCAP-31-03 5.2 – Embed lambda functions into the code** - lambdas: defining and using lambdas - self-defined functions taking lambdas as arguments - functions: map(), filter()	
			Hour 4	**PCAP-31-03 5.3 – Define and use closures** - closures: meaning and rationale - defining and using closures	
		Day 5	**Hour 1-4**	**PCAP-31-03 5.4 – Understand basic Input/output terminology** - I/O modes - predefined streams - handles vs. streams - text vs. binary modes	
Week 4		**Day 3**	**Hour 1-3**	**PCAP-31-03 5.5 – Perform Input/output operations** - the open() function - the errno variable and its values	**Task 9** *Details may be seen at Annexure-I*

				• functions: close(), .read(), .write(), .readline(), readlines() • using bytearray as input/output buffer	
			Hour 4	**Written Assessment** ***MCQ Type having 30 questions with maximum 4 options per question (Time Allowed: 40 Minutes)***	
	Get Certification	**Day 4 & 5**	**Hour 1-4**	**Exam Preparation and Online Registration** **Certified Associate in Python Programming (Exam PCAP-31-03)**	

AZ-204: Developing Solutions for Microsoft Azure

Week 5	**Azure Compute Solutions**	**Day 1**	**Hours #1-2**	• Introduction to Azure compute solutions	**Task 10** *Details may be seen at Annexure-I*
			Hour 3	• Implementing IaaS solutions	
			Hour 4	• Provisioning virtual machines (VMs)	
		Day 2	**Hour # 1-2**	• Configuring, validating, and deploying ARM templates	

			Hour # 3-4	• Configuring container images for solutions	
		Day 3	**Hour # 1-2**	• Publishing an image to Azure Container Registry	
			Hour # 3-4	• Running containers by using Azure Container Instance	
		Day 4	**Hour # 1-2**	• Creating an Azure App Service Web App	
			Hour #3	• Enabling diagnostics logging	
			Hour #4	• Deploying code to a web app	
		Day 5	**Hour # 1-2**	• Configuring web app settings including SSL, API settings, and connection strings	
			Hour # 2-3	• Implementing autoscaling rules including scheduled	

				autoscaling and autoscaling by operational or system metrics	
Week 6	**Azure Functions &** **Azure Storage**	**Day 1**	**Hour # 1-2**	**Introduction to Azure Functions** • Creating and deploying Azure Functions apps	**Task 11** *Details may be seen at Annexure-I*
			Hour # 3-4	• Implementing input and output bindings for a function	
		Day 2	**Hour # 1-4**	• Implementing function triggers by using data operations, timers, and webhooks	
		Day 3	**Hour # 1-4**	• Implementing Azure Durable Functions	
		Day 4	**Hour # 1-2**	**Introduction to Azure Storage** ● Developing solutions that use Azure Cosmos DB storage	
			Hour #3	• Selecting the appropriate API	

				and SDK for a solution	
			Hour #4	● Implementing partitioning schemes and partition keys	
		Day 5	**Hour #1-2**	● Performing operations on data and Azure Cosmos DB containers	
			Hour #3	● Setting the appropriate consistency level for operations	
			Hour #4	● Managing change feed notifications	
Week 7	**Azure Storage** **&** **Azure Security and Monitoring**	**Day 1**	**Hour #1-2**	**Developing solutions that use Blob Storage** ● Moving items in Blob storage between storage accounts or containers	**Task 12** *Details may be seen at Annexure-I*

			Hour #2-3	• Setting and retrieving properties and metadata	
			Hour #4	• Implementing storage policies, data archiving, and retention	
		Day 2	**Hour #1-3**	**Implementing user authentication and authorization** • Authenticating and authorizing users by using the Microsoft Identity platform	
			Hour #4	• **Authenticating and authorizing users and apps by using Azure AD**	
		Day 3	**Hour #1-2**	• Creating and implementing shared access signatures	
			Hour #3-4	• Implementing solutions that interact with Microsoft Graph	

		Day 4	**Hour # 1-2**	**Implementing secure cloud solutions**	
			Hour # 3-4	• Securing app configuration data by using App Configuration or Azure Key Vault	
		Day 5	**Hour #1-2**	• Developing code that uses keys, secrets, and certificates stored in Azure Key Vault	
			Hour # 3-4	• Implementing Managed Identities for Azure resources	
Week 8	**Azure Security and Monitoring**	**Day 1**	**Hour # 1**	**Monitoring, troubleshooting, and optimizing Azure solutions** • Implementing caching for solutions	**Task 13** *Details may be seen at Annexure-I*
			Hour # 2	• Configuring cache and expiration policies for Azure Cache for Redis	

			Hour # 3-4	● Implementing secure and optimized application cache patterns including data sizing, connections, encryption, and expiration	
		Day 2	**Hour # 1**	● Troubleshooting solutions by using metrics and log data	
			Hour # 2	● Configuring an app or service to use Application Insights	
			Hour # 3	● Reviewing and analyzing metrics and log data	
			Hour 4	● Implementing Application Insights web tests and alerts	

		Day 3	**Hour #1**	**Connecting to and consuming Azure services and third-party services** 1. Implementing API Management • Creating an APIM instance	
			Hour #2	• Creating and documenting APIs	
			Hour #3	• Configuring authentication for APIs	
			Hour #4	• Defining policies for APIs	
		Day 4	**Hour #1-2**	2. Developing event-based solutions • Implementing solutions that use Azure Event Grid	
			Hour #3	• Implementing solutions that use Azure Event Hub	
			Hour # 4	3. Developing message-based solutions • Implementing solutions that use Azure Service Bus	

		Day 5	**Hour # 1-2**	• Implementing solutions that use Azure Queue Storage queues	
			Hour #3-4	***MCQ Type having 30 questions with maximum 4 options per question (Time Allowed: 40 Minutes)*** ***F&Q***	
SAACO3 (AWS Certified Solutions Architect - Associate)					
Week 9	**Introduction to the AWS Certified Solutions Architect**	**Day 1**	**Hour # 1**	● Introduction to AWS Certification	**Task 14** *Details may be seen at Annexure-I*
			Hour # 2	● Introduction to AWS Certified Solutions Architect	
			Hour # 3	● Overview of AWS Certified Solutions Architect	
			Hour # 4	● Exam guide for AWS Certified Solutions Architect	
		Day 2	**Hour # 1**	● Introduction to exam-style questions	

			Hour # 2	● AWS Certified Solutions Architect - Associate Official Practice Question Set	
			Hour # 3	● AWS Certified Solutions Architect - Associate Sample Questions	
			Hour # 4	● AWS training suggestions ,Whitepapers and FAQs	
	Design Secure Architectures - Part 1	**Day 3**	**Hour #1-2**	● Introduction	
			Hour #3-4	● Access controls and management across multiple accounts	
		Day 4	**Hour # 1-2**	● AWS federated access and identity services (for example, AWS Identity and Access	
			Hour #3-4	● Management [IAM], AWS Single Sign-On [AWS SSO])	
		Day 5	**Hour # 1-4**	● AWS global infrastructure (for example, Availability	

				Zones, AWS Regions)	
				● AWS security best practices (for example, the principle of least privilege)	
Week 10	**Design Secure Architectures – Part 2**	**Day 1**	**Hour # 1-2**	● AWS security best practices (for example, the principle of least privilege) ● The AWS shared responsibility mode	**Task 15** *Details may be seen at Annexure-I*
			Hour # 3-4	***MCQ Type having 30 questions with maximum 4 options per question (Time Allowed: 40 Minutes)*** ***F&Q***	
	Design Resilient Architectures	**Day 2**	**Hour# 1**	• Introduction	**Task 16** *Details may be seen at Annexure-I*
			Hour# 2	• API creation and management (for example, Amazon API Gateway, REST API)	
			Hour # 3-4	• AWS managed services with appropriate use cases (for example, AWS Transfer Family, Amazon	

			Hour 4	• Simple Queue Service [Amazon SQS], Secrets Manager)	
		Day 3	**Hour # 1-2**	• Design principles for microservices (for example, stateless workloads compared with stateful workloads)	
			Hour # 3-4	• Event-driven architectures	
		Day 4	**Hour # 1-2**	• Queuing and messaging concepts (for example, publish/subscribe)	
			Hour # 3-4	• Serverless technologies and patterns (for example, AWS Fargate, AWS Lambda)	
		Day 5	**Hour #1-2**	• Storage types with associated characteristics (for example, object, file, block)	
			Hour #3-4	• The orchestration of containers (for example, Amazon Elastic Container Service [Amazon ECS], Amazon Elastic Kubernetes Service [Amazon EKS])	

Week 11	**Design High-Performing Architectures**	**Day 1**	**Hour #1**	● Introduction	**Task 17** *Details may be seen at Annexure-I*
			Hour # 2	● Hybrid storage solutions to meet business requirements	
			Hour # 3	● Storage services with appropriate use cases (for example, Amazon S3, Amazon Elastic File	
			Hour #4	● System [Amazon EFS], Amazon Elastic Block Store [Amazon EBS])	
		Day 2	**Hour #1**	● Storage types with associated characteristics (for example, object, file, block)	
			Hour # 2	● AWS compute services with appropriate use cases (for example, AWS Batch, Amazon EMR, Fargate)	
			Hour #3	● Distributed computing concepts supported by AWS global infrastructure and	

				edge services	
			Hour #4	● Queuing and messaging concepts (for example, publish/subscribe)	
		Day 3	**Hour #1**	● Distributed computing concepts supported by AWS global infrastructure and edge services	
			Hour #2	• Scalability capabilities with appropriate use cases (for example, Amazon EC2 Auto Scaling,	
			Hour# 3-4	• AWS Auto Scaling)	
		Day 4	**Hour #1**	• Serverless technologies and patterns (for example, Lambda, Fargate)	
			Hour# 2	• Data analytics and visualization services with appropriate use cases (for example, Amazon	
			Hour #3	• Athena, AWS Lake Formation, Amazon QuickSight)	
			Hour #4	• Data ingestion patterns (for example, frequency)	

		Day 5	**Hour #1**	• Data transfer services with appropriate use cases (for example, AWS DataSync, AWS Storage Gateway)	
			Hour #2	• Building and securing data lakes	
			Hour # 3	• Designing data streaming architectures	
			Hour #4	• Designing data transfer solutions	
Week 12	**Design Cost-Optimized Architectures**	**Day 1**	**Hour # 1**	• Introduction	**Task 18** *Details may be seen at Annexure-I*
			Hour #2	• Access options (for example, an S3 bucket with Requester Pays object storage)	
			Hour # 3-4	• AWS cost management service features (for example, cost allocation tags, multi-account billing)	
		Day 2	**Hour # 1**	• AWS cost management tools with appropriate use cases (for example, AWS Cost Explorer, AWS Budgets, AWS Cost and Usage Report)	

			Hour # 2-3	• AWS storage services with appropriate use cases (for example, Amazon FSx, Amazon EFS, Amazon S3, Amazon EBS)	
			Hour # 4	• Backup strategies	
		Day 3	**Hour # 1**	• Block storage options (for example, hard disk drive [HDD] volume types, solid state drive [SSD] volume types)	
			Hour #2	• Data lifecycles	
			Hour #3	• Hybrid storage options (for example, DataSync, Transfer Family, Storage Gateway)	
			Hour #4	• Storage access patterns	
		Day 4	**Hour #1**	• AWS cost management service features (for example, cost allocation tags, multi-account billing)	

			Hour # 2-3	• AWS cost management tools with appropriate use cases (for example, Cost Explorer, AWS Budgets, AWS Cost and Usage Report)	
			Hour #4	• AWS global infrastructure (for example, Availability Zones, AWS Regions)	
		Day 5	**Hour #1**	• AWS purchasing options (for example, Spot Instances, Reserved Instances, Savings Plans)	
			Hour # 2	• Budgets, AWS Cost and Usage Report)	
			Hour # 3	• Caching strategies	
			Hour #4	• Data retention policies	

Annexure-I

Tasks For Certificate in Software Development (PYTHON / JAVA / SQL / C++)

Task No.	Task	Description	Week No
Task 1	**Import and use modules packages and exceptions**	**Import and use modules and packages** 1. Write a Python program to import the NumPy package and create a 2D array. 2. Write a Python program to import the pandas module and create a DataFrame. 3. Write a Python program to import the matplotlib module and plot a line graph. 4. Write a Python program to import the seaborn module and plot a scatter plot. 5. Write a Python program to import the datetime module and display the current date and time. 6. Write a Python program to import the math module and calculate the value of pi. 7. Write a Python program to import the random module and generate a random number between 1 and 100. 8. Write a Python program to import the os module and list all the files in a directory. 9. Write a Python program to import the re module and search for a pattern in a string.	**Week–1**

		10. Write a Python program to import the requests module and send a GET request to a website.	
Task 2	**Strings**	**String Methods Tasks;** 1. Write a Python program to create a list of strings and use the .join() method to concatenate them with a separator. 2. Write a Python program to create a string and use the .split() method to split it into a list of words. 3. Write a Python program to create a list of strings and use the .sort() method to sort it in alphabetical order. 4. Write a Python program to create a list of strings and use the sorted() function to sort it in alphabetical order. 5. Write a Python program to create a string and use the .index() method to find the index of a substring. 6. Write a Python program to create a string and use the .find() method to find the first occurrence of a substring. 7. Write a Python program to create a string and use the .rfind() method to find the last occurrence of a substring. 8. Write a Python program to create a string and use the .upper() method to convert it to uppercase. 9. Write a Python program to create a string and use the .lower() method to convert it to lowercase.	**Week-2**
Task 3	**basic of oop (object oriented programing)**	**Perform any 5 tasks that involve the basics of OOP (Object Oriented Programming)** 1. Write a Python program to create a class called "Person" with attributes such as name, age, and address. 2. Write a Python program to create an object of the "Person" class and print the name of the person. 3. Write a Python program to create a class called "Rectangle" with attributes such as length and width, and methods such as "calculate_area" and	**Week-2**

		"calculate_perimeter". 4. Write a Python program to create an object of the "Rectangle" class and calculate the area and perimeter of the rectangle. 5. Write a Python program to create a class called "BankAccount" with attributes such as account number, balance, and methods such as "deposit" and "withdraw". 6. Write a Python program to create an object of the "BankAccount" class and deposit and withdraw money from the account. 7. Write a Python program to create a class called "Animal" with attributes such as name and species, and methods such as "make_sound". 8. Write a Python program to create a class called "Dog" that inherits from the "Animal" class and has additional attributes such as breed and methods such as "bark". 9. Write a Python program to create an object of the "Dog" class and make it bark. 10. Write a Python program to create a class called "Car" with attributes such as make, model, and year, and methods such as "start_engine" and "stop_engine". 11. Write a Python program to create an object of the "Car" class and start and stop the engine of the car.	
Task 4	**Component of Oop (Object Oriented programing)**	**Build a class hierarchy using inheritance** **Cover the topic With Examples** 1. Write a Python program to create a class called "Vehicle" with attributes such as speed and color, and a method called "move". 2. Write a Python program to create a class called "Car" that inherits from the "Vehicle" class and has additional attributes such as make and model. 3. Write a Python program to create a class called "Square"	**Week–3**

that inherits from both the "Rectangle" and "Circle" classes and has additional attributes such as side length.

4. Write a Python program to create objects of the "Rectangle", "Circle", and "Square" classes and call their "draw" methods.

5. Write a Python program to create a class called "Penguin" that inherits from both the "Animal" and "Bird" classes.

6. Write a Python program to create a class called "Manager" that inherits from the "Employee" class and has additional attributes such as bonus.

7. Write a Python program to create an object of the "Student" class and use the isinstance() function to check if it is an instance of the "Person" class.

8. Write a Python program to create a class called "Vector" with attributes such as x and y, and methods such as "add" and "subtract" that overload the "+" and "-" operators.

9. Write a Python program to create objects of the "Vector" class and use the overloaded "+" and "-" operators to add and subtract them.

10. Write a Python program to create a class called "Animal" with a method called "speak".

11. Write a Python program to create a class called "Cat" that inherits from the "Animal" class and overrides the "speak" method to print "meow".

12. Write a Python program to create a class called "Dog" that inherits from the "Animal" class and overrides the "speak" method to print "bark".

13. Write a Python program to create a list of objects of the "Animal", "Cat", and "Dog" classes and call their "speak" methods to demonstrate polymorphism.

14. Write a Python program to create a class called "Point" with attributes such as x and y, and methods such as "distance_to" that overload the "<" and ">" operators.

		15. Write a Python program to create objects of the "Point" class and use the overloaded "<" and ">" operators to compare their distances to the origin.	
Task 5	**Comprehension Taks**	**Comprehension Taks** 1. Write a Python program to create a list of common elements between two lists using list comprehension. 2. Write a Python program to create a list of tuples containing the elements of two lists using list comprehension. 3. Write a Python program to create a list of dictionaries from a list of tuples using list comprehension. 4. Write a Python program to create a list of all the prime numbers between 1 and 100 using list comprehension. **5.** Write a Python program to create a list of all the palindromic numbers between 1 and 100 using list comprehension.	**Week–4**
Task 6	**embedded lambda function**	**Perform any 5 Tasks on embedded lambda function** 1. Write a Python program to create a list of numbers and use an embedded lambda function to square each element. 2. Write a Python program to create a list of strings and use an embedded lambda function to convert each string to uppercase. 3. Write a Python program to create a dictionary of key-value pairs and use an embedded lambda function to sort it by value.	**Week–4**

		4. Write a Python program to create a list of tuples and use an embedded lambda function to sort it by the second element of each tuple. 5. Write a Python program to create a list of numbers and use an embedded lambda function to filter out all the even numbers. 6. Write a Python program to create a list of strings and use an embedded lambda function to filter out all the strings that start with a vowel. 7. Write a Python program to create a list of tuples and use an embedded lambda function to filter out all the tuples that have a sum greater than 10. 8. Write a Python program to create a list of dictionaries and use an embedded lambda function to filter out all the dictionaries that have a key 'name' with a value of 'John'. 9. Write a Python program to create a list of numbers and use an embedded lambda function to find the maximum value. 10. Write a Python program to create a list of tuples and use an embedded lambda function to find the tuple with the minimum value of the second element.	
Task7	**Tasks on Closure**	**Perform any 5 Tasks on Closure** 1. Write a Python program that defines a closure to calculate the average of a series of numbers. The closure should maintain a running total of the numbers and the count of the numbers seen so far. 2. Write a Python program that defines a closure to generate a sequence of Fibonacci numbers. The closure should keep track of the last two numbers	**Week–4**

		generated and return the next number in the sequence each time it is called. 3. Write a Python program that defines a closure to create a function that doubles a number. The closure should take a multiplier argument that sets the factor by which the number is doubled. 4. Write a Python program that defines a closure to create a function that calculates the nth power of a number. The closure should take an exponent argument that specifies the power to which the number should be raised. 5. Write a Python program that defines a closure to create a function that returns the nth prime number. The closure should keep track of the primes seen so far and only test odd numbers for primality after the first two primes (2 and 3).	
Task 8	**Tasks on I/O Operations**	**Perform any 5 Tasks on I/O Operations** 1. Write a Python program to read a text file and print its contents. 2. Write a Python program to read a CSV file and print its contents. 3. Write a Python program to read a JSON file and print its contents. 4. Write a Python program to write a text file with some text. 5. Write a Python program to write a CSV file with some data. 6. Write a Python program to write a JSON file with some data. 7. Write a Python program to read an XML file and print	**Week–4**

		its contents. 8. Write a Python program to write an XML file with some data. 9. Write a Python program to read an Excel file and print its contents. 10. Write a Python program to write an Excel file with some data. 11. Write a Python program to read a binary file and print its contents. 12. Write a Python program to write a binary file with some data. 13. Write a Python program to copy the contents of one file to another. 14. Write a Python program to append some text to an existing text file.	
Task 9	**Certification**	**Exam Preparation and Online Registration** **Certified Associate in Python Programming (Exam PCAP-31-03)** **Take Exam and Pass Certification**	**Week 4**

Task 10	**Azure Web App Deployment and Scaling**	**Task:** Create an Azure Web App and deploy a sample web application to it. Configure the app's SSL, API settings, and connection strings. Implement autoscaling rules, including scheduled autoscaling and autoscaling by operational or system metrics. **Deliverables:** A report detailing the process of creating and deploying the web app, including screenshots of configuration settings and a description of the autoscaling rules implemented.	**Week 5**
Task 11	**Azure Functions Development and Deployment**	**Task:** Create and deploy an Azure Function app that uses input and output bindings. Implement function triggers using data operations, timers, and webhooks. Use Azure Durable Functions to orchestrate multiple functions in a workflow. **Deliverables:** A report detailing the process of creating and deploying the	**Week 6**

		function app, including screenshots of configuration settings and a description of the implemented triggers and orchestrations.	

Task 12	**Azure Cosmos DB Implementation**	**Task:** Develop a solution that uses Azure Cosmos DB storage. Select the appropriate API and SDK for the solution, implement partitioning schemes and partition keys, and perform operations on data and containers. Set the appropriate consistency level for operations and manage change feed notifications. **Deliverables**: A report detailing the process of implementing the solution, including screenshots of configuration settings and a description of the implemented partitioning schemes and consistency level. Demonstrate the use of change feed notifications.	**Week 7**

Task 13	**Azure API Management Implementation**	**Task:** Create an Azure API Management instance and document APIs. Configure authentication for APIs and define policies. Use Azure Service Bus to implement a message-based solution and Azure Event Grid to implement an event-based solution. **Deliverables:** A report detailing the process of creating the API Management instance, including screenshots of configuration settings and a description of the implemented authentication and policies. Demonstrate the use of the message-based and event-based solutions.	**Week 8**

Task 14	**Design secure access to AWS resources Part 1**	• Apply AWS security best practices to IAM users and root users (for example, multi-factor • authentication [MFA]) • Design a flexible authorization model that includes IAM users, groups, roles, and policies • Design a role-based access control strategy (for example, AWS Security Token Service [AWS • STS], role switching, cross-account access) • Design a security strategy for multiple AWS accounts (for example, AWS Control Tower, • service control policies [SCPs])	**Week–9**
Task 15	**Design secure access to AWS resources. – Part 2**	• Determine the appropriate use of resource policies for AWS services • Determine when to federate a directory service with IAM roles	**Week-10**
Task 16	**Design scalable and loosely coupled architectures**	• Design event-driven, microservice, and/or multi-tier architectures based on requirements • Determine scaling strategies for components used in an architecture design • Determine the AWS services required to achieve loose coupling based on requirements • Determine when to use containers • Determine when to use serverless technologies and patterns	**Week-10**

Task 17	**Design high-performing and elastic compute solutions**	• Determine storage services and configurations that meet performance demands • Determine storage services that can scale to accommodate future needs • Decouple workloads so that components can scale independently • Identify metrics and conditions to perform scaling actions • Build and securing data lakes • Design data streaming architectures	**Week-11**
Task 18	**Design cost-optimized storage solutions**	• Design appropriate storage strategies (for example, batch uploads to Amazon S3 compared • with individual uploads) • Determine the correct storage size for a workload • Determine the lowest cost method of transferring data for a workload to AWS storage • Determine when storage auto scaling is required • Managing S3 object lifecycles • Select the appropriate backup and/or archival solution • Design appropriate backup and retention policies (for example, snapshot frequency) • Migrating database schemas and data to different locations and/or different database engines • Selecting the appropriate bandwidth allocation for a network device (for example, a single VPN compared with multiple VPNs, Direct Connect speed)	**Week-12**

Annexure-II

SUGGESTIVE FORMAT AND SEQUENCE ORDER OF MOTIVATIONAL LECTURE.

Mentor

Mentors are provided an observation checklist form to evaluate and share their observational feedback on how students within each team engage and collaborate in a learning environment. The checklist is provided at two different points: Once towards the end of the course. The checklists are an opportunity for mentors to share their unique perspective on group dynamics based on various team activities, gameplay sessions, pitch preparation, and other sessions, giving insights on the nature of communication and teamwork taking place and how both learning outcomes and the student experience can be improved in the future.

Session- 1 (Communication):

Please find below an overview of the activities taking place Session plan that will support your delivery and an overview of this session's activity.

Session- 1 OVERVIEW
Aims and Objectives:
· To introduce the communication skills and how it will work · Get to know mentor and team - build rapport and develop a strong sense of a team · Provide an introduction to communication skills · Team to collaborate on an activity sheet developing their communication, teamwork, and problem-solving · Gain an understanding of participants' own communication skills rating at the start of the program

Activity:	**Participant Time**	**Teacher Time**	**Mentor Time**
Intro Attend and contribute to the scheduled.			
Understand good communication skills and how it works.			
Understand what good communication skills mean			
Understand what skills are important for good communication skills			

Key learning outcomes:	**Resources:**	**Enterprise skills developed:**
· Understand the communication skills and how it works. · Understand what communication skills mean · Understand what skills are important for	· Podium · Projector · Computer · Flip Chart · Marker	· Communication · Self Confidence · Teamwork

communication skills		

Schedule	Mentor Should do
Welcome: **5 min**	Short welcome and ask the **Mentor** to introduce him/herself. Provide a brief welcome to the qualification for the class. Note for Instructor: Throughout this session, please monitor the session to ensure nothing inappropriate is being happened.
Icebreaker: **10 min**	Start your session by delivering an icebreaker, this will enable you and your team to start to build rapport and create a team presentation for the tasks ahead. The icebreaker below should work well at introductions and encouraging communication, but feel free to use others if you think they are more appropriate. It is important to encourage young people to get to know each other and build strong team links during the first hour; this will help to increase their motivation and communication throughout the sessions.
Introduction & Onboarding: **20mins**	Provide a brief introduction of the qualification to the class and play the “Onboarding Video or Presentation”. In your introduction cover the following: 1. Explanation of the program and structure. (Kamyab jawan Program) 2. How you will use your communication skills in your professional life. 3. Key contacts and key information – e.g. role of teacher, mentor, and SEED. Policies and procedures (user agreements and “contact us” section). Everyone to go to the Group Rules tab at the top of their screen, read out the rules, and ask everyone to verbally agree. Ensure that the consequences are clear for using the platform outside of hours. (9am-8pm)

	4. What is up next for the next 2 weeks ahead so young people know what to expect (see pages 5-7 for an overview of the challenge). Allow young people to ask any questions about the session topic.
Team Activity Planning: **30 minutes**	MENTOR: Explain to the whole team that you will now be planning how to collaborate for the first and second collaborative Team Activities that will take place outside of the session. There will not be another session until the next session so this step is required because communicating and making decisions outside of a session requires a different strategy that must be agreed upon so that everyone knows what they are doing for this activity and how. · “IDENTIFY ENTREPRENEURS” TEAM ACTIVITY · “BRAINSTORMING SOCIAL PROBLEMS” TEAM ACTIVITY” *As a team, collaborate on a creative brainstorm on social problems in your community. Vote on the areas you feel most passionate about as a team, then write down what change you would like to see happen.* Make sure the teams have the opportunity to talk about how they want to work as a team through the activities e.g. when they want to complete the activities, how to communicate, the role of the project manager, etc. Make sure you allocate each young person a specific week that they are the project manager for the weekly activities and make a note of this. Type up notes for their strategy if this is helpful - it can be included underneath the Team Contract.

Session Close: **5 minutes**	**MENTOR:** Close the session with the opportunity for anyone to ask any remaining questions. **Instructor:** Facilitate the wrap-up of the session. A quick reminder of what is coming up next and when the next session will be.

MOTIVATIONAL LECTURES LINKS.

TOPIC	**SPEAKER**	**LINK**
How to Face Problems In Life	**Qasim Ali Shah**	**https://www.youtube.com/watch?v=OrQte08Ml90**
Just Control Your Emotions	**Qasim Ali Shah**	**https://www.youtube.com/watch?v=JzFs__yJt-w**
How to Communicate Effectively	**Qasim Ali Shah**	**https://www.youtube.com/watch?v=PhHAQEGehKc**
Your ATTITUDE is Everything	**Tony Robbins Les Brown David Goggins Jocko Willink Wayne Dyer Eckart Tolle**	**https://www.youtube.com/watch?v=5fS3rj6eIFg**

Control Your EMOTIONS	**Jim Rohn** **Les Brown** **TD Jakes** **Tony Robbins**	**https://www.youtube.com/watch?v=chn86sH0O5U**
Defeat Fear, Build Confidence	**Shaykh Atif Ahmed**	**https://www.youtube.com/watch?v=s10dzfbozd4**
Wisdom of the Eagle	**Learn Kurooji**	**https://www.youtube.com/watch?v=bEU7V5rJTtw**
The Power of ATTITUDE	**Titan Man**	**https://www.youtube.com/watch?v=r8LJ5X2ejqU**
STOP WASTING TIME	**Arnold Schwarzenegger**	**https://www.youtube.com/watch?v=kzSBrJmXqdg**
Risk of Success	**Denzel Washington**	**https://www.youtube.com/watch?v=tbnzAVRZ9Xc**

Annexure-III

Suggestive Format and Sequence Order of Success Story

S. No	Key Information	Detail/Description
1.	**Self & Family background**	• Self-introduction • Family background and socio economic status, • Education level and activities involved in • Financial hardships etc
2.	**How he came on board NAVTTC Training/ or got trained through any other source**	• Information about course, apply and selection • Course duration, trade selection • Attendance, active participation, monthly tests, interest in lab work
3.	**Post training activities**	• How job / business (self-employment) was set up • How capital was managed (loan (if any) etc). • Detail of work to share i.e. where is job or business being done; how many people employed (in case of self-employment/ business) • Monthly income or earnings and support to family • Earning a happy life than before
4.	**Message to others (under training)**	• Take the training opportunity seriously • Impose self-discipline and ensure regularity

		• Make Hard work pays in the end so be always ready for the same.

<u>Note: Success story is a source of motivation for the trainees and can be presented in a number of ways/forms in a NAVTTC skill development course as under:-</u>

1. **To call a passed out successful person of institute. He/she will narrate his/her success story to the trainees in his/her own words and meet trainees as well.**
2. **To see and listen to a recorded video/clip (5 to 7 minutes) showing a successful person Audio video recording that has to cover the above mentioned points.**
3. **The teacher displays the picture of a successful trainee (name, trade, institute, organization, job, earning per month etc) and narrates his/her story in teacher's own motivational words.**

Annexure-IV

Workplace/Institute Ethics Guide

Work ethic is a standard of conduct and values for job performance. The modern definition of what constitutes good work ethics often varies. Different businesses have different expectations. Work ethic is a belief that hard work and diligence have a moral benefit and an inherent ability, virtue, or value to strengthen character and individual abilities. It is a set of values-centered on the importance of work and manifested by determination or desire to work hard.

The following ten work ethics are defined as essential for student success:

1. **<u>Attendance</u>:**

   Be at work every day possible, plan your absences don't abuse leave time. Be punctual every day.
2. **<u>Character</u>:**

   Honesty is the single most important factor having a direct bearing on the final success of an individual, corporation, or product. Complete assigned tasks correctly and promptly. Look to improve your skills.
3. **<u>Team Work:</u>**

The ability to get along with others including those you don’t necessarily like. The ability to carry your weight and help others who are struggling. Recognize when to speak up with an idea and when to compromise by blend ideas together.

4. **Appearance**:

   Dress for success set your best foot forward, personal hygiene, good manner, remember that the first impression of who you are can last a lifetime

5. **Attitude**:

   Listen to suggestions and be positive, accept responsibility. If you make a mistake, admit it. Values workplace safety rules and precautions for personal and co-worker safety. Avoids unnecessary risks. Willing to learn new processes, systems, and procedures in light of changing responsibilities.

6. **Productivity**:

   Do the work correctly, quality and timelines are prized. Get along with fellows, cooperation is the key to productivity. Help out whenever asked, do extra without being asked. Take pride in your work, do things the best you know-how. Eagerly focuses energy on accomplishing tasks, also referred to as demonstrating ownership. Takes pride in work.

7. **Organizational Skills**:

   Make an effort to improve, learn ways to better yourself. Time management; utilize time and resources to get the most out of both. Take an appropriate approach to social interactions at work. Maintains focus on work responsibilities.

8. **Communication**:

   Written communication, being able to correctly write reports and memos. Verbal communications, being able to communicate one on one or to a group.

9. **Cooperation**:

   Follow institute rules and regulations, learn and follow expectations. Get along with fellows, cooperation is the key to productivity. Able to welcome and adapt to changing work situations and the application of new or different skills.

10. **Respect**:

    Work hard, work to the best of your ability. Carry out orders, do what’s asked the first time. Show respect, accept, and acknowledge an individual’s talents and knowledge. Respects diversity in the workplace, including showing due respect for different perspectives, opinions, and suggestions.