

Government of Pakistan

National Vocational and Technical Training Commission

Prime Minister's Youth Skills Development Program



Course Contents / Lesson Plan

Course Title: ORACLE JAVA + ANGULAR

Duration: 3 Months

Revised Edition

Trainer Name	
Course Title	ORACLE JAVA + ANGULAR
Objectives and Expectations	<p>This specialized course is crafted to address the demand for skilled professionals in Oracle Java and Angular, enhancing employability through hands-on practice and practical skills. The primary goal is to move beyond theoretical knowledge and equip participants with the capabilities to function effectively in the job market either independently or as part of a team. The curriculum is designed to deliver not only technical expertise in Oracle Java and Angular but also to emphasize the development of soft skills (such as interpersonal and communication skills, personal presentation) and entrepreneurial skills (including marketing and freelancing). Furthermore, the course aims to instill a strong work ethic to promote better professional conduct and enhance the reputation of the Pakistani workforce.</p> <p>Main Expectations: The key expectation is that, under the guidance of experienced professionals, participants will gain proficiency to apply Oracle Java and Angular skills for earning income upon completion. This approach underscores a market-centric strategy, necessitating instructors to understand market roles, assess individual strengths and weaknesses, and prepare participants for relevant positions.</p> <p>Implementation: Specially designed practical tasks related to Oracle Java and Angular are included in Annexure-I, with records maintained for monitoring visits. A dedicated module on Job Search & Entrepreneurial Skills in the 5th & 6th month addresses local and international job markets, visa processes, and opportunities for self-employment.</p> <p>A module on Workplace Ethics emphasizes positive behavior in professional settings, contributing to an enhanced image of the Pakistani workforce. Maintaining Interest: To keep participants motivated, modern techniques like motivational lectures, success stories, and case studies are employed. These tools, detailed in the training methodology, aim to sustain interest and objectively assess acquired competencies.</p> <ul style="list-style-type: none"> (i) Motivational Lectures: (ii) Motivational lectures, a crucial component, aim to inspire participants towards professional excellence in Oracle Java and Angular. These lectures include a clear purpose, personal stories, participant relevance, and compelling ending points to drive creativity and curiosity. (iii) Success Stories: (iv) Success stories, shared through various mediums, motivate participants by showcasing the journey to success in Oracle Java and Angular. The training institute is expected to regularly present high-quality success stories.

	<p>(iii) Case Studies: Where suitable, case studies enhance understanding by presenting real-life scenarios in Oracle Java and Angular. This method encourages participants to actively analyze and discuss practical challenges related to Oracle Java and Angular. The ultimate objective is to not only impart technical knowledge but also transform participants into responsible and skilled professionals ready to excel in the Oracle Java and Angular job market.</p>
Entry-level of trainees	<p>For an advanced Java and Angular course, the suggested entry requirement is a minimum of a bachelor's degree in a related field. Thus, participants are expected to:</p> <ul style="list-style-type: none"> • Possess a solid understanding of Programming Concepts. • Have previous exposure to programming languages such as C, C++, and Python. • Demonstrate a comprehension of computer systems. • Exhibit a foundational knowledge of basic electronics.
Learning Outcomes of the course	<ul style="list-style-type: none"> • Upon completion of the Java and Angular course, participants will emerge with advanced programming skills in Java and Angular, enabling them to design and develop dynamic web applications. • They will gain expertise in utilizing Java libraries, implementing design patterns, integrating databases, and conducting both client-side (Angular) and server-side (Java) development. • The course emphasizes full-stack development, covering testing, debugging, version control, collaboration, and project management. Participants will acquire hands-on experience through real-world projects, building a comprehensive portfolio that reflects their proficiency in Java and Angular. • Overall, the course equips participants with the knowledge and skills necessary for impactful contributions to web and software development projects in various industries.
Course Execution Plan	<p>The total duration of the course: 3 months (12 Weeks) Class hours: 4 hours per day Theory: 20% Practical: 80% Weekly hours: 20 hours per week Total contact hours: 240 hours Java SE 17 Developer Certification</p>
Companies offering jobs in the respective trade	<ul style="list-style-type: none"> • Oracle Corporation: Naturally, Oracle itself is a major player in Java development. They offer various positions for Java developers to work on Java SE, Java EE, and related technologies. • IBM: IBM is known for its involvement in Java development, especially with its WebSphere application server and other enterprise solutions. • Microsoft: While Microsoft is not primarily associated with Oracle Java, they have been increasingly incorporating Java into their technologies, such as Azure services and development tools.

	<ul style="list-style-type: none"> • Google: Google uses Java extensively in its projects, including Android development. Google Cloud Platform (GCP) also provides opportunities for Java developers. • Amazon (AWS): Amazon Web Services (AWS) supports Java, and there are opportunities for Java developers to work on various cloud-related projects. • Red Hat: As a leading provider of open-source solutions, Red Hat has positions for Java developers, especially those with experience in Java EE and related technologies. • Capgemini: This global consulting and technology services company often has job openings for Java developers. • Accenture: Accenture is a multinational professional services company that frequently hires Java developers for various projects. • Infosys: Infosys is a global IT consulting and services company that often has positions for Java developers in its workforce. • Tata Consultancy Services (TCS): TCS is one of the largest IT services companies globally, and they regularly hire Java developers for their projects. • Pakistan Petroleum Limited (PPL) • Oil & Gas Development Company Limited (OGDCL) • Pakistan Telecommunication Company Limited (PTCL) • Jazz (formerly Mobilink and Warid) • Habib Bank Limited (HBL) • MCB Bank Limited • United Bank Limited (UBL) • Allied Bank Limited (ABL) • Engro Corporation • Packages Limited • Millat Tractors Limited •
Job Opportunities	<ul style="list-style-type: none"> • Software Engineer • Web Developer • Database Administrator • IT Project Manager • Network Engineer • Customer Service Representative • Banking Officer • Financial Analyst • Relationship Manager • Petroleum Engineer • Geologist • Mechanical Engineer • Electrical Engineer
No of Students	25
Learning Place	Classroom / Lab
Instructional Resources	<p>Oracle Java:</p> <ul style="list-style-type: none"> • Oracle Java Tutorials:

- Java Tutorials on Oracle's Official Website
- Java Programming and Software Engineering Fundamentals (Coursera):
- Codecademy - Learn Java:
 - Codecademy Java Course
- Java Brains - Java Programming Tutorials:
 - Java Brains YouTube Channel
- Baeldung - Java Programming Guides:
 - Baeldung Java Guides

AngularJS:

- AngularJS Official Documentation:
- AngularJS Documentation
- AngularJS Tutorial for Beginners (edX):
 - edX AngularJS Course
- Angular - The Complete Guide (Udemy):
 - Angular - The Complete Guide on Udemy
 - Coursera - Single Page Web Applications with AngularJS:
 - Coursera AngularJS Course
- AngularJS Video Tutorials by Academind:
 - Academind AngularJS Playlist
 - W3Schools - AngularJS Tutorial:
 - W3Schools AngularJS Tutorial

MODULES

Schedu led Weeks	Module Title	Days	Hours	Learning Units	Home Assignment	
Week 1	Java Basics	Day 1	Hour 1	An introduction to the Java programming language, its history, and significance.	Task -1 <i><u>Details may be seen at Annexure-1</u></i>	
				Overview of Java History and evolution of Java JDK, JRE, and JVM		
			Hour 2	Understanding the syntax rules and structure of Java code.		Task -2
			Hour 3	Outputting information to the console using various methods in Java.		Task -3
			Hour 4	Exploring different types of comments and their role in code documentation.		Task -4
	Data, Variables and Operators	Day 2	Hour (1-4)	Introduction to variables, their declaration, and usage in Java programs.	Task -5	
	Day 3	Hour (1-2)	Exploring different data types such as int, double, char, boolean, etc.	Task -6		
	Day 4	Hour (1-4)	Converting one data type to another in Java.	Task -7		
	Day 5	Hour (1-4)	Overview of operators in Java, including arithmetic, logical, and relational.	Task -8		

Week 2	Strings, Math, Datatypes and Control Flow and Loops	Day 1	Hour (1-4)	Working with strings in Java, including string manipulation and methods.	Task -9		
		Day 2	Hour (1-4)	Utilizing mathematical operations and functions available in Java.	Task -10		
		Day 3	Hour (1-4)	Understanding boolean data type and logical operations in Java.	Task -11		
		Day 4	Hour (1-2)	Implementing conditional statements based on certain conditions.	Task -12		
			Hour (3-4)	Using switch statements for multi-way branching in Java.	Task -13		
		Day 5	Hour (1-2)	Introduction to the while loop for repetitive execution of code.	Task -14		
			Hour (3-4)	Implementing loops with a fixed number of iterations using for loops.	Task -15		
			Hour (1-4)	Understanding the use of break and continue statements in loops.	Task -16		
		Week 3	Arrays and Methods and Scope and Recursion	Day 1	Hour (1-2)	Understanding arrays and their usage to store multiple values in Java.	Task -17
					Hour (3-4)	Introduction to methods/functions and their role in modularizing code.	Task -18
Day 2	Hour (1-2)			Passing parameters to methods for enhanced functionality.	Task -19		
	Hour (3-4)			Defining multiple methods with the same name but different parameters.	Task -20		

		Day 3	Hour (1-4)	Understanding variable scope and its impact on code structure.	Task -21
		Day 4	Hour (1-4)	Implementing recursive methods, where a method calls itself.	Task -22,23
		Day 5	Hour (1-4)	Revision and Practice the previous Topics	
Week 4	Java OOP	Day 1	Hour (1-2)	Introduction to classes as the fundamental building blocks of Java programs.	Task (24-32)
			Hour (3-4)	Overview of Object-Oriented Programming (OOP) concepts in Java.	
		Day 2	Hour (1-2)	Creating and using objects from classes in Java.	
			Hour (3-4)	Defining attributes and fields within Java classes.	
		Day 3	Hour (1-2)	Implementing methods within Java classes for encapsulation of behavior.	
			Hour (3-4)	Introduction to constructors for initializing objects in Java.	
		Day 4	Hour (1-2)	Understanding access modifiers (public, private, protected) in Java.	
			Hour (3-4)	Utilizing encapsulation to protect the internal state of objects.	
		Packages and Inheritance	Day 5	Hour (3-4)	

			Hour (1-4)	Understanding the concept of inheritance and its implementation in Java.	
Week 5	Polymorphism and Inner Classes	Day 1	Hour (1-2)	Implementing polymorphism for multiple forms of a method in Java.	Task 33-43
			Hour (2-4)	Introduction to inner classes and their use within Java programs.	
	Advanced Java Features	Day 2	Hour (1-2)	Implementing abstraction for hiding implementation details in Java.	
			Hour (3-4)	Creating and implementing interfaces in Java.	
		Day 3	Hour (1-2)	Introduction to enumeration types in Java.	
			Hour 3	Accepting user input in Java applications.	
			Hour 4	Working with date and time in Java.	
		Java Collections and Advanced Topics	Day 4	Hour (1-2)	
	Hour (3-4)			Understanding linked lists and their usage in Java.	
	Day 5		Hour (1-2)	Working with hash maps for key-value storage in Java.	
			Hour (3-4)	Introduction to hash sets and their applications in Java.	
	Week 6	Java Collections and Advanced Topics	Day 1	Hour (1-2)	
Hour (3-4)				Working with wrapper classes for primitive data types.	
Swing Framework		Day 2	Hour (1-2)	Handling exceptions and errors in Java.	
			Hour (3-4)	Utilizing regular expressions for	

				pattern matching in Java.	
		Day 3	Hour (1-2)	Working with files and file I/O operations in Java.	Task 48
			Hour (3-4)	Swing Framework: Understanding Swing components (JFrame, JPanel, JButton, etc.). Layout managers for arranging components. Event handling in Swing. JavaFX:	Task 49, 50
		Day 4	Hour (1-2)	Introduction to JavaFX and its advantages over Swing. Building user interfaces using FXML and Scene Builder. Event handling in JavaFX.	
			Hour (3-4)	Java GUI Components: Text components (JTextField, JTextArea). List components (JList, JTable). Combo boxes and other input components.	Task 51
		Day 5	Hour (1-2)	Layout Management: Understanding layout managers (FlowLayout, BorderLayout, GridLayout, etc.). Custom layout managers.	Task 52, 53

				<p>Graphics and Custom Painting:</p> <p>Drawing shapes and images. Custom painting on components. Internationalization and Accessibility:</p>	
			Hour (3-4)	<p>Localizing applications for different languages. Making applications accessible. Multithreading in UI:</p> <p>SwingWorker for background tasks. Event dispatch thread (EDT). Drag-and-Drop:</p> <p>Implementing drag-and-drop functionality. Data Binding:</p> <p>Connecting UI components to data models.</p>	Task 54
Week 7	Styling and Themes for JAVA UI Components JavaFX Framework	Day 1	Hour (1-2)	<p>Styling and Themes:</p> <p>Applying styles and themes to UI components. Java Web Start:</p> <p>Deploying Java applications using Java Web Start.</p>	Task 55
			Hour (3-4)	<p>Swing vs. JavaFX:</p> <p>Understanding the differences and choosing the right framework.</p>	
			Hour		

		Day 2	(1-2)	Sealed Classes and Records	Task 56-59
			Hour (3-4)	Exploring the new features introduced in Java SE 17	
		Day 3	Hour (1-2)	Pattern Matching and Switch Expressions	
			Hour (3-4)	Understanding pattern matching and switch expressions	
		Day 4	Hour (1-4)	Applying these features in Java programming	
		Day 5	Hour (1-4)	Revision of JAVA UI Components JavaFX Framework	
Week 8	Java SE 17 Features	Day 1	Hour (1-4)	Comprehensive Review	
		Day 2	Hour (1-4)	Recap of key Java SE 17 concepts and features	
		Day 3	Hour (1-4)	Engaging in practice tests to assess understanding	
		Day 4	Hour (1-4)	Strategies for taking the Java SE 17 Developer Certification exam	
		Day 5	Hour (1-4)	Strategies for taking the Java SE 17 Developer Certification exam	
Week 9	Appear in Java SE 17 Developer Certification exam	Day 1-5	Hour (1-20)	Appear in Java SE 17 Developer Certification exam	
Week 10	Introduction to Angular Basics	Day 1	Hour (1-4)	Overview of Angular, its usage, and environment setup	Task 60-64
		Day 2	Hour (1-4)	Templates and Data Binding Interpolation and Property Binding	
		Day 3	Hour (1-4)	Angular Services and HTTP Client Dependency Injection, Creating	

				and Consuming Services	
		Day 4	Hour (1-4)	Angular Forms Basics Template-Driven Forms, Reactive Forms	
		Day 5	Hour (1-4)	Angular Routing and Navigation Nested Routes and Route Guards, Lazy Loading Modules	
Week 11	Advanced Angular Concepts and Best Practices	Day 1	Hour (1-4)	Angular Modules and Pipes Feature Modules, Shared Modules, Angular Pipes	Task 64-69
		Day 2	Hour (1-4)	Angular Testing Unit Testing with Jasmine and Karma, Testing Components and Services	
		Day 3	Hour (1-4)	Angular Animations and Material Design Basics of Animations, Angular Material Introduction	
		Day 4	Hour (1-4)	NgRx State Management NgRx Basics, Actions, Reducers, Selectors, Handling Asynchronous Operations	
		Day 5	Hour (1-4)	Angular Security and PWA Security Best Practices, XSS and CSRF Prevention	
Week 12	Create Freelancing Proficiency on Digital Platforms	Day 1	Hour 1 to 4	Introduction to Freelancing on Digital Platforms Overview of digital freelancing: Explanation of various platforms (Upwork, Fiverr, Freelancer, etc.), types of projects available, and skill	<ul style="list-style-type: none"> Task 70 <i>Details may be seen at Annexure-1</i>

				<p>requirements.</p> <p>Understanding the benefits, challenges, and strategies for success in the digital freelancing landscape.</p>	
		Day 2	Hour 1 to 4	<p>Profile Creation and Optimization</p> <p>Guide on setting up an effective freelancer profile: Creating a compelling bio, selecting skills, showcasing portfolios, and highlighting expertise.</p> <p>Optimizing profiles on popular freelancing platforms with relevant keywords and best practices.</p>	
		Day 3	Hour 1 to 4	<p>Understanding Client Communication and Proposals</p> <p>Exploring effective client communication: Strategies for professional interaction, understanding client needs, and crafting compelling proposals.</p> <p>Conducting role plays or mock scenarios to practice effective</p>	

				communication and proposal writing.	
--	--	--	--	-------------------------------------	--

Tasks for Certificate in Oracle Java + Angular

Task No.	Task	Description	Week
1	Install JDK and IDE	This task involves installing the Java Development Kit (JDK) and an Integrated Development Environment (IDE) such as Eclipse, IntelliJ, or NetBeans. The JDK provides the necessary tools for Java development, while the IDE offers a user-friendly environment for writing, compiling, and managing Java code.	Week 1
2	Write and run a simple Java program	Create a basic Java program that includes a <code>main</code> method. Write code to print a simple message to the console. Compile and execute the program to verify that it runs without errors.	Week 1
3	Create a program to display output messages	Develop a Java program that displays various output messages using the <code>System.out.println()</code> statement. Experiment with different types of messages, including text and variables.	Week 1
4	Add comments to the Java program for documentation	Enhance the readability of your code by adding comments. Include both single-line and multi-line comments to describe the purpose of the program, important methods, and any complex logic.	Week 1
5	Declare and use variables in a program	Declare variables of different data types (<code>int</code> , <code>double</code> , <code>String</code> , etc.) and demonstrate their usage within the program. Assign values, perform operations, and print the results.	Week 1
6	Create a program that utilizes different data types	Develop a Java program that involves variables of various data types, such as primitive types (<code>int</code> , <code>float</code>) and reference types (<code>String</code>). Illustrate the use of each data type in different scenarios.	Week 1
7	Perform type casting operations in a Java program	Implement type casting operations to convert data from one type to another. Demonstrate both implicit and explicit type casting in the context of your program.	Week 1
8	Write a program that uses various operators in Java	Create a program that utilizes arithmetic, relational, logical, and bitwise operators. Illustrate how each operator works and influences the program's behavior.	Week 1
9	Create a program to manipulate and display strings in Java	Develop a Java program that manipulates strings. Use string concatenation, substring operations, and other string manipulation techniques. Display the results to demonstrate the changes.	Week 2
10	Develop a program that performs mathematical calculations in Java	Write a Java program that includes mathematical calculations. Use operators for addition, subtraction, multiplication, and division. Incorporate variables to store and display the results.	Week 2
11	Implement a program using boolean data type and logical operations	Create a program that involves boolean variables and logical operations. Use conditions to control the flow of the program based on boolean values.	Week 2
12	Write a program with conditional statements in Java	Utilize <code>if</code> , <code>else if</code> , and <code>else</code> statements to implement conditional logic in your program. Demonstrate the execution of different code blocks based on specified conditions.	Week 2

13	Create a program that utilizes switch statements	Implement a Java program with switch statements to handle multiple cases efficiently. Use this control structure to make decisions based on different values of a variable.	Week 2
14	Develop a program with a while loop for repetitive tasks	Utilize a while loop to perform repetitive tasks in your program. Demonstrate how the loop continues execution as long as a specified condition is true.	Week 2
15	Write a program that uses for loop for iteration	Implement a for loop to iterate over a sequence of values. Show how the loop facilitates repetitive tasks and controls the number of iterations.	Week 2
16	Modify a program using break and continue statements	Enhance your program by incorporating break and continue statements within loops. Illustrate how these statements affect the flow of the loop.	Week 2
17	Create a program that utilizes arrays in Java	Develop a Java program that declares, initializes, and manipulates arrays. Showcase the use of single-dimensional and multi-dimensional arrays.	Week 3
18	Define and call methods in a Java program	Introduce methods into your program. Define reusable blocks of code, call them from the main program, and observe how methods enhance code organization.	Week 3
19	Modify a program to include method parameters	Extend your program by adding method parameters. Explore the flexibility of passing different types of parameters to methods.	Week 3
20	Implement method overloading in a Java program	Showcase method overloading by creating multiple methods with the same name but different parameter lists. Demonstrate how the appropriate method is invoked based on the provided arguments.	Week 3
21	Create a program to demonstrate variable scope in Java	Illustrate variable scope by declaring variables in different scopes (local, instance, class). Demonstrate how scope affects the visibility and lifetime of variables.	Week 3
22	Develop a recursive program in Java	Implement a recursive function within your program. Demonstrate how the function calls itself to solve a problem, emphasizing the termination condition.	Week 3
23	Create a simple class in Java	Introduce the concept of classes by creating a basic class. Include attributes and methods within the class.	Week 4
24	Implement basic OOP principles in a Java program	Incorporate basic Object-Oriented Programming (OOP) principles, such as encapsulation, inheritance, and polymorphism, in your program.	Week 4
25	Develop a program that utilizes objects in Java	Instantiate objects of classes and utilize them within your program. Demonstrate how objects encapsulate data and behavior.	Week 4
26	Modify a program to include class attributes	Extend your program by adding attributes to the class. Illustrate how objects can have different states based on their attribute values.	Week 4
27	Define and use methods within a class in Java	Expand the functionality of your class by defining and using methods within it. Showcase how methods can manipulate class attributes.	Week 4

28	Implement constructors in a Java program	Introduce constructors to initialize class attributes when objects are created. Demonstrate the use of default and parameterized constructors.	Week 4
29	Modify a program to include access modifiers	Enhance your program by incorporating access modifiers (public, private, protected) to control the visibility of class members.	Week 4
30	Implement encapsulation in a Java program	Emphasize encapsulation by making class attributes private and providing public methods for accessing and modifying them.	Week 4
31	Structure a program using packages in Java	Organize your program into packages to manage and structure code more efficiently. Demonstrate how to import classes from other packages.	Week 4
32	Implement inheritance in a Java program	Create a program that utilizes inheritance. Establish a hierarchy of classes to showcase the reuse of attributes and methods.	Week 4
33	Create a program that demonstrates polymorphism in Java	Implement polymorphism by showcasing methods with the same signature in different classes. Illustrate how method overriding enables dynamic method invocation.	Week 5
34	Develop a program that utilizes inner classes in Java	Create inner classes within your program. Demonstrate how inner classes have access to the outer class's members.	Week 5
35	Create an abstract class and use it in a program	Define an abstract class and extend it within your program. Demonstrate how abstract classes provide a blueprint for concrete classes.	Week 5
36	Implement interfaces in a Java program	Introduce interfaces into your program. Showcase how classes can implement multiple interfaces to achieve multiple inheritances.	Week 5
37	Utilize enum types in a Java program	Utilize enum types in a Java program	Week 5
38	Develop a program that takes user input in Java	Develop a program that takes user input in Java	Week 5
39	Create a program that uses date and time functions in Java	Create a program that uses date and time functions in Java	Week 5
40	Implement a program that uses ArrayList in Java	Implement a program that uses ArrayList in Java	Week 5
41	Develop a program that utilizes LinkedList in Java	Develop a program that utilizes LinkedList in Java	Week 5
42	Create a program that uses HashMap in Java	Create a program that uses HashMap in Java	Week 5

43	Implement a program that uses HashSet in Java	Implement a program that uses HashSet in Java	Week 5
44	Develop a program that uses iterators in Java	Develop a program that uses iterators in Java	Week 6
45	Implement a program that uses wrapper classes in Java	Implement a program that uses wrapper classes in Java	Week 6
46	Create a program that demonstrates exception handling in Java	Create a program that demonstrates exception handling in Java	Week 6
47	Develop a program that uses regular expressions in Java	Develop a program that uses regular expressions in Java	Week 6
48	Implement a program that reads, writes, and manipulates files in Java	Implement a program that reads, writes, and manipulates files in Java	Week 6
49	Create and Implement Layout using Swing Framework	<p>Understanding Swing Components:</p> <p>Create a simple JFrame with buttons, labels, and text fields. Implement a JPanel containing various Swing components like checkboxes and radio buttons. Customize the appearance of Swing components using different styles and sizes.</p> <p>Layout Managers:</p> <p>Experiment with different layout managers (FlowLayout, BorderLayout, GridLayout) to arrange components. Create a complex GUI with nested layouts to understand the hierarchical arrangement of components.</p> <p>Event Handling in Swing:</p> <p>Implement ActionListener and(ActionEvent) for button clicks. Use ItemListener for handling events on checkboxes and radio buttons. Explore MouseListener for mouse-related events on Swing components.</p>	Week 6
50	Create and Implement Layout using JavaFX Framework	<p>Introduction to JavaFX:</p> <p>Create a basic JavaFX application with a stage and scene. Use JavaFX components such as Button, Label, and TextField. Apply CSS styling to JavaFX components for a visually appealing interface.</p> <p>FXML and Scene Builder:</p> <p>Build a JavaFX interface using FXML and Scene Builder.</p>	Week 6

		<p>Link FXML elements to corresponding controller methods. Incorporate CSS styles and themes using FXML.</p> <p>Event Handling in JavaFX:</p> <p>Implement EventHandler for handling button clicks and other UI events.</p> <p>Explore the MouseEvent class for mouse-related events in JavaFX.</p>	
51	Create UI Components using Java GUI Components	<p>Text Components:</p> <p>Develop a Java application with JTextArea for multiline text input.</p> <p>Implement JTextField for single-line text input.</p> <p>List Components:</p> <p>Create a JList with a list of items that trigger specific actions.</p> <p>Implement a JTable for displaying tabular data.</p> <p>Combo Boxes and Input Components:</p> <p>Utilize JComboBox for a dropdown selection.</p> <p>Implement various input components like sliders and spinners.</p>	Week 7
52	Create Layouts using Layout Manager	<p>Layout Management:</p> <p>Understanding Layout Managers:</p> <p>Create a GUI application using FlowLayout, BorderLayout, and GridLayout.</p> <p>Experiment with the GridBagConstraints in GridBagLayout.</p> <p>Custom Layout Managers:</p> <p>Develop a custom layout manager for a specific application layout.</p> <p>Apply the custom layout manager to a JFrame containing multiple components.</p>	Week 7
53	Usage of Graphics and Custom Painting	<p>Graphics and Custom Painting:</p> <p>Drawing Shapes and Images:</p> <p>Draw basic shapes (circles, rectangles) on a JPanel.</p> <p>Load and display images on a JFrame.</p> <p>Custom Painting:</p> <p>Implement custom painting on a JPanel using the paintComponent method.</p> <p>Create an interactive drawing application with custom shapes.</p>	Week 7
54	Internationalization and Accessibility	<p>Internationalization and Accessibility:</p> <p>Localizing Applications:</p> <p>Implement localization by providing translations for different languages.</p> <p>Create resource bundles for internationalization.</p> <p>Accessibility:</p>	Week 7

		Implement accessibility features such as screen readers and keyboard navigation. Ensure the application complies with accessibility standards.	
55	Apply Styling and Themes	Styling and Themes: Applying Styles and Themes: Apply CSS styles to JavaFX components. Change the overall theme of the JavaFX application using predefined styles. Java Web Start: Deploying Java Applications using Java Web Start: Package a Java application for deployment using Java Web Start. Create a simple web page to launch the Java Web Start application.	Week 7
56	Implement sealed classes and records in Java SE 17.	Implement sealed classes and records in Java SE 17. Apply them to enhance code clarity and security.	Week 8
57	Use pattern matching and switch expressions in solving practical programming problems.	Use pattern matching and switch expressions in solving practical programming problems. Refactor existing code to leverage these new features.	Week 8
58	Review and revise previous Java UI components, incorporating the latest Java SE 17 features.	Review and revise previous Java UI components, incorporating the latest Java SE 17 features. Ensure compatibility and optimize code using the new language enhancements.	Week 8
59	Extend JavaFX applications to incorporate the latest Java SE 17 features.	Extend JavaFX applications to incorporate the latest Java SE 17 features. Explore how sealed classes and records can enhance the JavaFX framework.	Week 8
60	Set up Angular Development Environment	Set up Angular Development Environment	Week 10
61	Build a Simple Angular Component	Build a Simple Angular Component	Week 10
62	Make API Requests in Angular	Make API Requests in Angular	Week 10
63	Handle Form Submissions	Handle Form Submissions	Week 10
64	Build a Multi-Page Angular Application	Build a Multi-Page Angular Application	Week 10
65	Create Custom Pipes and Directives	Create Custom Pipes and Directives	Week 11

66	Implement Unit Tests for Angular Components	Implement Unit Tests for Angular Components	Week 11
67	Build UI with Angular Material Components	Build UI with Angular Material Components	Week 11
68	Implement NgRx in Angular Application	Implement NgRx in Angular Application	Week 11
69	Build a Progressive Web App (PWA) with Angular	Build a Progressive Web App (PWA) with Angular	Week 11
70	Create Freelancing Proficiency on Digital Platforms	Develop proficiency in digital freelancing by covering profile optimization, client communication, project management tools, financial aspects, and reputation building on popular platforms in a week-long course.	Week 12

Motivational Lectures Oracle Java + Angular Js

Oracle Java Certification | OCA Crash Course:

https://www.youtube.com/watch?v=IWKIVLAIGnc&list=PLviC8AFqAj5BEtYhyh2QUgUHjzIU_61Ib

Videos in the OCA and OCP Certification preparation series, this video covers the details about the exam, some tips and strategy how to pass the exam.

Oracle Java Certification | Book for Core Java

You can refer to the below book for core java: <https://www.pdfdrive.com/java-the-com...>

Angular Js

<https://www.youtube.com/@Angular>

About

Angular an application development platform that lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop. For more info, visit <http://angular.io>

Channel details

www.youtube.com/@Angular

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. Attendance:

Be at work every day possible, plan your absences don't abuse leave time. Be punctual every day.

2. Character:

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. Team Work:

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.