

<b>COURSE TITLE</b>	<b>FUNDAMENTALS OF APPLICATION DEVELOPMENT</b>
<b>COURSE CODE</b>	<b>05MB0101</b>
<b>COURSE CREDITS</b>	<b>4</b>

**Objective:**

- 1 Understand Java's history, features, OOP principles, inheritance, polymorphism, packages, and access specifiers for proficient Java development.
- 2 Arrays, sorting, strings, string buffer, and exception handling including predefined and user-defined exceptions.
- 3 Learn GUI components, event handling, layouts, and implement interfaces for interactive Java applications.
- 4 Master Swing components like JMenu, JTabbedPane, JTable, JScrollPane, and Swing listeners for creating feature-rich and interactive desktop applications.
- 5 Explore JDBC architecture, driver types, statements, and result sets for effective Java database connectivity.

**Course Outcomes:** After completion of this course, student will be able to:

- 1 Gain proficiency in Java development through a deep understanding of its history, features, OOP principles, inheritance, polymorphism, packages, and access specifiers.
- 2 Master arrays, sorting techniques, string manipulation, string buffer usage, and robust exception handling including predefined and user-defined exceptions for efficient Java programming.
- 3 Acquire skills in GUI components, event handling, layout design, and interface implementation to create interactive and user-friendly Java applications.
- 4 Proficiently utilize Swing components like JMenu, JTabbedPane, JTable, JScrollPane, and Swing listeners to develop feature-rich and interactive desktop applications.
- 5 Gain a thorough understanding of JDBC architecture, various driver types, SQL statements, and result sets to establish efficient Java database connectivity in applications.

**Pre-requisite of course:** Basic Knowledge of Programming

**Teaching and Examination Scheme**

<b>Theory Hours</b>	<b>Tutorial Hours</b>	<b>Practical Hours</b>	<b>ESE</b>	<b>IA</b>	<b>CSE</b>	<b>Viva</b>	<b>Term Work</b>
2	0	4	50	30	20	0	50

<b>Contents : Unit</b>	<b>Topics</b>	<b>Contact Hours</b>
1	<b>Java Introduction</b> Why Java , Paradigms , Diff B/W Java & Other (C,C++) , Java History, Java Features , Java programming format , Java Statements , Java Data Types, Introduction of OOPS , Object , Constructors , This Key Word ,Inheritance , Super Key Word, Polymorphism (Over Loading & Over Riding) , Abstraction , Interface , Encapsulation, Introduction to all predefined packages , User Defined Packages , Access Specifiers, What is Array , Single Dimensional Array ,Multi-Dimensional Array , Sorting of Arrays, String ,String Buffer, Introduction of Exception Handling , Pre-Defined Exceptions , Try-Catch-Finally , Throws, throw , User Defined Exception examples	15
2	<b>AWT, Swing, Database &amp; Result set</b> Introduction, Components, Event-Delegation-Model, Listeners, Layouts, Individual Components Label, Button, Check Box, Radio Button, Choice, List, Menu, Text Field, Text Area, Event handling using interfaces. Examples of event handling on Button, Checkbox, Choices, List, Menu, Swing Components- JMenu, JTabbedPane, JTable, JScrollPane, Swing Listeners, Introduction, JDBC Architecture , Types of Drivers, Type-1, Type-2, Type-3, Type-4, Statement, Result Set, Read Only Result Set, Updatable Result Set	15
<b>Total Hours</b>		<b>30</b>

**Suggested List of Experiments:**

<b>Contents : Unit</b>	<b>Topics</b>	<b>Contact Hours</b>
1	<b>Unit 1</b> Write a simple Java program to print "Hello, World!" to the console., Create a Java program with variables, data types, and basic arithmetic operations., Implement a Java program using OOP concepts such as classes, objects, methods, and instance variables, Demonstrate Java control statements like if-else, switch-case, loops (for, while, do-while), and their usage in practical scenarios, Create a hierarchy of classes demonstrating inheritance in Java., In your java program Use the 'super' keyword to access superclass members and constructors., Design a Java program that uses encapsulation principles by defining private instance variables and public getter/setter methods, Explore Java's built-in packages like java.util, java.io, and java.lang by using their classes and methods in practical programs., Create a custom Java package with multiple classes and demonstrate package-level access control., Create a Java program to initialize and manipulate a single-dimensional array, such as finding the sum or average of elements., Implement a multi-dimensional array for storing and processing data, like a matrix or a table in java program., Write Java code to sort a single-dimensional array using different algorithms like bubble sort, selection sort, or merge sort., Extend the sorting functionality to multi-dimensional arrays for sorting rows or columns in your java program. , Develop a Java program to perform string manipulation operations like concatenation, substring extraction, and searching for specific characters or substrings., Create a Java program that demonstrates the use of try-catch blocks to handle exceptions such as arithmetic exceptions or null pointer exceptions.	30

**Suggested List of Experiments:**

<b>Contents : Unit</b>	<b>Topics</b>	<b>Contact Hours</b>
2	<b>Unit 2</b> Create a simple Java GUI application with basic components like labels, buttons, checkboxes, and text fields., Implement event listeners (e.g., ActionListener, ItemListener) to handle user interactions with GUI components in java program., Implement interfaces like ActionListener, ItemListener, or MouseListener to handle events for specific components in java program., Create a java practical examples of event handling on buttons, checkboxes, choice menus, lists, and menu items., Create a java practical in which use event handling to perform actions like displaying messages, updating GUI components, or processing user input., Create a Java Swing application with a menu bar containing multiple menus and menu items., Implement event listeners (ActionListener or ItemListener) to handle menu item selections and perform actions accordingly., Write a java program in which add components (e.g., labels, buttons, text fields) to each tab and demonstrate switching between tabs., Populate a JTable with data from a data source (e.g., an array, list, or database) in java program., Create a scrollable view for components that exceed the visible area, such as large text areas or tables in java program., Create a java practical examples of event handling on Swing components like buttons, checkboxes, and tabs using listeners., Write a Java program to establish a connection with a database using JDBC., Implement a java code examples for each type of driver to connect to a database and perform basic CRUD operations., Create a java program in which use JDBC statements (Statement, PreparedStatement, CallableStatement) to execute SQL queries and updates., Write a java program in which fetch and process data from a database using JDBC result sets (ResultSet)., Create a java practical to Demonstrate the usage of read-only result sets and updatable result sets for different database operations.	30
<b>Total Hours</b>		<b>60</b>

**Textbook :**

- 1 Java: A Beginner's Guide, Herbert Schildt, McGraw-Hill Education, 2018

**References:**

- 1 Java Programming: From Problem Analysis to Program Design, Java Programming: From Problem Analysis to Program Design, D. S. Malik, Cengage Learning, 2018
- 2 Java: The Complete Reference, Java: The Complete Reference, Herbert Schildt, McGraw-Hill Education, 2021
- 3 Java Concurrency in Practice, Java Concurrency in Practice, Brian Goetz et al., Addison-Wesley, 2006

**Suggested Theory Distribution:**

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery					
<b>Remember / Knowledge</b>	<b>Understand</b>	<b>Apply</b>	<b>Analyze</b>	<b>Evaluate</b>	<b>Higher order Thinking / Creative</b>
10.00	20.00	25.00	25.00	10.00	10.00

**Instructional Method:**

- 1 Board Work
- 2 PPT
- 3 Demo

**Supplementary Resources:**

- 1 <https://www.javatpoint.com/java-tutorial>
- 2 <https://docs.oracle.com/javase/tutorial/>
- 3 <https://www.tutorialspoint.com/java/index.htm>