UNIT I – OOP Basics, Data Types, Operators & Control Statements

Q: What are the main principles of OOP?

A: Encapsulation, Inheritance, Polymorphism, and Abstraction.

Q: Define encapsulation and abstraction.

A: Encapsulation hides data; Abstraction shows essential features only.

Q: What is the structure of a simple Java program?

A: It includes a class, main method, statements, and closing braces.

Q: What is the difference between JDK, JRE, and JVM?

A: JDK is for development, JRE for execution, JVM runs bytecode.

Q: What are tokens in Java?

A: Smallest program elements like keywords, literals, operators.

Q: What is the purpose of main() method?

A: It's the entry point for program execution.

Q: What are primitive data types?

A: byte, short, int, long, float, double, char, boolean.

Q: What is type casting?

A: Converting one data type to another.

Q: What is operator precedence?

A: Defines order of operator evaluation.

Q: What is the difference between while and do-while?

A: While checks before executing; do-while executes once at least.

Q: What are command-line arguments?

A: Values passed to main() via command line.

Q: Difference between if-else and switch?

A: if-else for conditions; switch for fixed values.

UNIT II – Classes, Objects, and Methods

Q: What is a class?

A: A blueprint defining attributes and behaviors.

Q: What is an object?

A: An instance of a class created using new.

Q: What is a constructor?

A: Special method used to initialize objects.

Q: What is constructor overloading?

A: Multiple constructors with different parameters.

Q: What is method overloading?

A: Same name, different parameters.

Q: What is method overriding?

A: Subclass redefines superclass method.

Q: What is the 'this' keyword?

A: Refers to current class instance.

Q: What are static variables?

A: Shared among all instances of a class.

Q: What is recursion?

A: A method calling itself repeatedly.

Q: What is method nesting?

A: Calling one method inside another.

Q: Can we overload main()?

A: Yes, but JVM calls standard main(String[] args).

Q: What are access modifiers?

A: Control visibility: public, private, protected, default.

UNIT III – Arrays, Inheritance, and Interfaces

Q: How do you declare an array?

A: int[] a = new int[5];

Q: What is inheritance?

A: Acquiring properties from another class.

Q: Types of inheritance?

A: Single, Multilevel, Hierarchical.

Q: What is the super keyword?

A: Refers to parent class members.

Q: What is method overriding?

A: Redefining parent method in subclass.

Q: What is dynamic method dispatch?

A: Runtime polymorphism via overridden methods.

Q: What is an abstract class?

A: Cannot be instantiated, may have abstract methods.

Q: What is an interface?

A: Collection of abstract methods.

Q: What are default methods?

A: Methods with body inside interfaces (Java 8+).

Q: What is multiple inheritance via interfaces?

A: A class implementing multiple interfaces.

UNIT IV – Packages, Exception Handling, and I/O

Q: What is a package?

A: Collection of related classes and interfaces.

Q: How to create a package?

A: Use 'package packagename;' at file top.

Q: How to import a package?

A: Using 'import packagename.*;'

Q: What are wrapper classes?

A: Convert primitives into objects.

Q: What is autoboxing?

A: Automatic conversion primitive→object.

Q: What is unboxing?

A: Automatic conversion object→primitive.

Q: What is an exception?

A: An event disrupting normal program flow.

Q: What are checked exceptions?

A: Handled at compile time.

Q: What are unchecked exceptions?

A: Handled at runtime.

Q: What is try-catch-finally?

A: Handles exceptions and cleanup code.

Q: What is throw vs throws?

A: throw = raise; throws = declare.

Q: What is File I/O?

A: Reading/writing data to files using streams.

UNIT V – Strings, Multithreading, and JavaFX

Q: Difference between String, StringBuffer, StringBuilder?

A: String immutable, others mutable.

Q: Why are Strings immutable?

A: For security and thread safety.

Q: What is multithreading?

A: Running multiple parts of program simultaneously.

Q: How to create a thread?

A: Extend Thread or implement Runnable.

Q: What is the life cycle of a thread?

A: New → Runnable → Running → Blocked → Terminated.

Q: What is synchronization?

A: Controlling shared resource access.

Q: What is a deadlock?

A: Threads waiting forever for each other.

Q: What is join()?

A: Waits for a thread to finish.

Q: What is a daemon thread?

A: Background service thread.

Q: What is JavaFX?

A: GUI toolkit for desktop apps.

Q: What are Stage and Scene?

A: Stage = window, Scene = content container.

Q: What is event handling?

A: Responding to user actions in UI.