Java Viva Questions and Answers

UNIT I - OOP Basics & Java Foundations

Q: What are the four main OOP principles?

A: Encapsulation, Inheritance, Polymorphism, and Abstraction.

Q: What is encapsulation?

A: It is the process of wrapping data (variables) and code (methods) into a single unit called class.

Q: What is inheritance?

A: Inheritance allows a class to inherit properties and methods from another class using the 'extends' keyword.

Q: What is polymorphism?

A: Polymorphism allows methods to perform different tasks based on the object calling them. It is of two types: compile-time (method overloading) and runtime (method overriding).

Q: What is abstraction?

A: Abstraction is the concept of hiding complex implementation details and showing only the required features.

Q: Difference between method overloading and overriding?

A: Overloading means same method name with different parameters in same class. Overriding means redefining a superclass method in subclass with same signature.

Q: What is the purpose of 'this' keyword?

A: It refers to the current object and is used to resolve naming conflicts.

Q: What is garbage collection?

A: Automatic memory management that removes unused objects to free up memory.

Q: Difference between local, instance and class variables?

A: Local variables are inside methods, instance variables are declared in class but outside methods, and class variables use 'static'.

Q: What are control statements?

A: Statements that control the flow of execution like if, switch, while, for, break, continue.

UNIT II - Inheritance, Packages, Interfaces

Q: What is hierarchical inheritance?

A: When multiple classes inherit from a single superclass.

Java Viva Questions and Answers

Q: What is substitutability?

A: A subclass object can be used wherever a superclass object is expected.

Q: What are the forms of inheritance in Java?

A: Specialization, Specification, Extension, Construction, Limitation, Combination.

Q: What is the use of 'super' keyword?

A: 'super' is used to refer to the immediate parent class object or constructor.

Q: What does 'final' keyword do in inheritance?

A: Prevents method overriding or class inheritance.

Q: What is polymorphism in Java?

A: It allows one interface to be used for a general class of actions.

Q: What is an abstract class?

A: A class that cannot be instantiated and can have abstract methods.

Q: What is the Object class?

A: The root class of all classes in Java.

Q: What is CLASSPATH?

A: Environment variable that tells Java where to look for user-defined classes and packages.

Q: Difference between class and interface?

A: Class can have both concrete and abstract methods; interface has only abstract methods (before Java 8).

UNIT III - Exception Handling and Multithreading

Q: What is exception handling?

A: Mechanism to handle runtime errors to maintain normal flow of application.

Q: What are checked and unchecked exceptions?

A: Checked are compile-time exceptions, unchecked are runtime exceptions.

Q: What is 'throw' vs 'throws'?

A: 'throw' is used to explicitly throw an exception, 'throws' is used to declare it.

Q: What is finally block?

A: It is always executed regardless of exception occurrence.

Q: What is the hierarchy of exceptions?

A: Throwable > Exception > RuntimeException.

Q: How to create a custom exception?

A: By extending Exception class and defining your own message.

Q: What is multithreading?

A: Concurrent execution of two or more threads to maximize CPU usage.

Q: Difference between multithreading and multitasking?

A: Multithreading runs multiple threads in one process, multitasking runs multiple processes.

Q: What are the thread states?

A: New, Runnable, Running, Blocked, Waiting, Timed Waiting, Terminated.

Q: What is synchronization?

A: Preventing thread interference when multiple threads access shared resources.

UNIT IV - Event Handling and AWT

Q: What is an event in Java?

A: A change in state generated by user interaction.

Q: What is event delegation model?

A: Design model where event handling is done by listener interfaces.

Q: What are adapter classes?

A: Classes with empty implementations of listener interfaces.

Q: Difference between KeyListener and MouseListener?

A: KeyListener handles keyboard events, MouseListener handles mouse actions.

Q: What are layout managers?

A: They automatically arrange components in containers. Types: BorderLayout, GridLayout, FlowLayout, CardLayout, GridBagLayout.

Q: Difference between Frame and Panel?

A: Frame is a top-level window, Panel is a generic container.

Q: What is Canvas used for?

A: It is a blank rectangular area for drawing graphics.

Q: How to handle a button click?

A: Use ActionListener and override actionPerformed().

Q: How to display event name on screen?

A: Override appropriate mouse event methods and use Graphics.drawString().

Q: What are text components?

A: AWT components like TextField, TextArea to input or display text.

UNIT V - Applets and Swing

Q: What is an applet?

A: A small Java program that runs in a browser or applet viewer.

Q: Applet vs Application?

A: Applets have no main(), applications do. Applets run in browser, applications run independently.

Q: Applet life cycle methods?

A: init(), start(), paint(), stop(), destroy().

Q: How to pass parameters to applets?

A: Using <PARAM> tag in HTML.

Q: What is Swing?

A: A GUI toolkit in Java with richer components than AWT.

Q: AWT vs Swing?

A: AWT is heavyweight and platform dependent; Swing is lightweight and platform independent.

Q: What is MVC architecture?

A: Model-View-Controller separates data (Model), UI (View), and interaction logic (Controller).

Q: What are containers in Swing?

A: Components that hold other components. Examples: JFrame, JPanel.

Q: What is JComponent?

A: Base class for all Swing components.

Q: What are some Swing components?

A: JButton, JLabel, JTextField, JCheckBox, JRadioButton, JComboBox, JTable, JTree.