**1. What is inheritance in Java?**

a) The process of creating multiple instances of a class

b) The process of hiding data and methods within a class

c) The process of reusing code from existing classes

d) The process of combining data and methods into a single unit

**Answer:**

c) The process of reusing code from existing classes

Explanation:

Inheritance is a mechanism in Java that allows a class to inherit properties and behaviors from another class. It promotes code reuse by enabling the creation of subclasses that inherit the attributes and methods of a superclass. Subclasses can also add their own unique attributes and methods.

**2. What is the difference between method overloading and method overriding in Java?**

a) Method overloading occurs within the same class, while method overriding occurs between different classes.

b) Method overloading involves creating multiple methods with the same name but different parameters, while method overriding involves providing a different implementation for an inherited method.

c) Method overloading is a compile-time polymorphism concept, while method overriding is a runtime polymorphism concept.

d) All of the above.

**Answer:**

d) All of the above.

Explanation:

All the statements are true. Method overloading allows the definition of multiple methods with the same name but different parameters within the same class. Method overriding occurs when a subclass provides a different implementation for a method that is already defined in its superclass. Method overloading is resolved at compile-time, while method overriding is resolved at runtime.

**3. What is polymorphism in Java?**

a) The ability of a class to inherit properties and behaviors from another class

b) The process of hiding data and methods within a class

c) The process of creating multiple instances of a class

d) The ability of an object to take on many forms

**Answer:**

d) The ability of an object to take on many forms

Explanation:

Polymorphism refers to the ability of an object to take on many forms or have multiple behaviors. In Java, polymorphism is achieved through method overriding and method overloading. It allows objects of different classes to be treated as objects of a common superclass, providing flexibility and extensibility.

**4. What are abstract classes in Java?**

a) Classes that cannot be instantiated

b) Classes that can be used as blueprints for creating objects

c) Classes that only contain abstract methods

d) All of the above

**Answer:**

d) All of the above

Explanation:

Abstract classes in Java cannot be instantiated directly and are typically used as blueprints for creating objects. They can contain abstract methods (methods without implementation) and regular methods. Abstract classes provide a way to define common behavior and enforce specific methods to be implemented by subclasses.

**5. What is the purpose of the "super" keyword in Java?**

a) To refer to the current object

b) To invoke the superclass constructor or methods

c) To create multiple instances of a class

d) To hide data and methods within a class

**Answer:**

b) To invoke the superclass constructor or methods

Explanation:

The "super" keyword in Java is used to refer to the superclass (or parent class) of the current object. It is commonly used to invoke the superclass constructor or methods within the subclass. The "super" keyword allows for code reuse and accessing superclass members that may be overridden in the subclass.

**6. What is the difference between a class and an object in Java?**

a) A class is a blueprint for creating objects, while an object is an instance of a class.

b) A class is a single entity, while an object is a collection of entities.

c) A class contains data and methods, while an object only contains data.

d) A class cannot be instantiated, while an object can be created and used.

**Answer:**

a) A class is a blueprint for creating objects, while an object is an instance of a class.

Explanation:

A class in Java is a template or blueprint that defines the structure and behavior of objects. It specifies the attributes (data) and methods that objects of that class will have. An object, on the other hand, is an instance of a class. It represents a specific entity or instance created based on the class blueprint.

**7. What is method overriding in Java?**

a) Creating multiple methods with the same name but different parameters within the same class.

b) Providing a different implementation for an inherited method in a subclass.

c) Hiding data and methods within a class.

d) Allowing a class to inherit properties and behaviors from another class.

**Answer:**

b) Providing a different implementation for an inherited method in a subclass.

Explanation:

Method overriding occurs when a subclass provides a different implementation for a method that is already defined in its superclass. The method in the subclass must have the same name, return type, and parameters as the superclass method. Method overriding allows for polymorphism and dynamic method dispatch.

**8. What is the purpose of the "final" keyword in Java?**

a) To prevent the inheritance of a class

b) To prevent overriding of a method

c) To prevent modification of a variable's value

d) All of the above

**Answer:**

d) All of the above

Explanation:

The "final" keyword in Java can be used to prevent the inheritance of a class, overriding of a method, or modification of a variable's value. When a class, method, or variable is declared as final, it cannot be further extended, overridden, or modified, respectively.

**9. What is the purpose of the "abstract" keyword in Java?**

a) To prevent the inheritance of a class

b) To prevent overriding of a method

c) To create an instance of a class

d) To declare an abstract class or method

**Answer:**

d) To declare an abstract class or method

Explanation:

The "abstract" keyword in Java is used to declare an abstract class or method. An abstract class cannot be instantiated and serves as a blueprint for creating derived classes. An abstract method does not have an implementation and must be overridden in the subclass.

**10. What is the difference between static and instance variables in Java?**

a) Static variables are associated with the class itself, while instance variables are associated with an instance of a class.

b) Static variables are shared among all instances of a class, while instance variables have separate values for each instance.

c) Static variables can be accessed without creating an object, while instance variables require an object reference.

d) All of the above.

**Answer:**

d) All of the above.