

30 Lecture - CS304

Important Mcqs

1. What is polymorphism?

- A) The ability of an object to take on many forms
- B) The ability of an object to change its type at runtime
- C) The ability of an object to inherit multiple classes
- D) The ability of an object to override its superclass methods

Answer: A

What is a common use case for polymorphism?

- A) Implementing different payment methods
- B) Creating complex mathematical algorithms
- C) Defining new data types
- D) All of the above

Answer: A

In a payroll application, how might polymorphism be used?

- A) To implement a common interface for calculating employee pay
- B) To ensure that all employees are paid the same amount
- C) To limit the types of employees that can be added to the system
- D) To prevent employees from accessing each other's pay information

Answer: A

What is an interface in Java?

- A) A concrete implementation of a class
- B) A template for defining a class
- C) A set of methods and constants that a class must implement
- D) A way to declare private methods in a class

Answer: C

What is method overloading?

- A) Defining multiple methods with the same name but different parameters
- B) Defining multiple methods with the same name and same parameters
- C) Overriding a superclass method with a subclass method
- D) None of the above

Answer: A

What is method overriding?

- A) Defining multiple methods with the same name but different parameters
- B) Defining multiple methods with the same name and same parameters
- C) Overriding a superclass method with a subclass method
- D) None of the above

Answer: C

What is the difference between method overloading and method overriding?

- A) Method overloading is done at compile time, while method overriding is done at runtime

- B) Method overloading is done by the superclass, while method overriding is done by the subclass
- C) Method overloading is used to define new methods, while method overriding is used to modify existing methods
- D) Method overloading is based on the number and type of parameters, while method overriding is based on the method name and parameters

Answer: D

Can a subclass access private methods and fields of its superclass?

- A) Yes, always
- B) No, never
- C) Only if the subclass is in the same package as the superclass
- D) Only if the superclass declares the methods and fields as protected

Answer: B

What is an abstract class in Java?

- A) A class that cannot be instantiated
- B) A class that does not have any methods
- C) A class that only has private methods
- D) A class that can only be used as a superclass

Answer: A

Can an abstract class have non-abstract methods?

- A) Yes, but only if the class also has at least one abstract method
- B) Yes, but only if the class has no abstract methods
- C) No, an abstract class can only have abstract methods
- D) None of the above

Answer: B