Q1: What is the expected output of the **main** method?

```java
class Parent {
    void printMessage() {
        System.out.println("Message from Parent");
    }
}

class Child extends Parent {
    @Override
    void printMessage() {
        System.out.println("Message from Child");
    }
}

public class InheritanceExample {
    public static void main(String[] args) {
        Parent parent = new Parent();
        Parent childAsParent = new Child();
        Child child = new Child();

        parent.printMessage();
        childAsParent.printMessage();
        child.printMessage();
    }
}
```

A )

```
Message from Parent
Message from Parent
Message from Child
```

b)

```
Message from Parent
Message from Child
Message from Child
```

c)

```
Message from Parent
Message from Child
Message from Parent
```

d)

```
Message from Child
Message from Child
Message from Child
```

## Ans:

**Q2:** Which of the following statements about interfaces in Java are true? "You can choose many options"

A) An interface can have constructors.

B) A class can implement multiple interfaces.

C) Interface methods are by default final and static.

D) An interface can extend multiple classes.

## Ans:.

**Q3**: Which of the following statements about polymorphism in Java are true? "You can choose many options."

A) Polymorphism allows a subclass to inherit the members of multiple super classes.

B) Method overloading is an example of compile-time polymorphism.

C) In Java, polymorphism is achieved through interfaces only.

D) Dynamic polymorphism is resolved during compile-time.

## Ans:

**Q:4** In the context of Java programming and object-oriented principles, what is encapsulation?

"You can choose many options."

A) The process of making an object's state publicly accessible through direct access to its fields.

B) A mechanism that restricts the access of some components of an object and prevents the accidental modification of its state.

C) The ability of a class to inherit multiple interfaces.

D) The process of bundling data and methods that operate on that data within a single unit.

**Ans:**

**Q4 :**

```
public class Shape {
    protected String color;

    public Shape(String color) {
        this.color = color;
    }

    public void draw() {
        System.out.println("Drawing a shape of color " + color);
    }
}
public interface Resizable {
    void resize(int percentage);
}
public class Circle extends Shape implements Resizable {
    private double radius;

    public Circle(String color, double radius) {
        super(color);
        this.radius = radius;
    }

    public void draw() {
        System.out.println("Drawing a circle of color " + color + " and radius "
+ radius);
    }

    public void resize(int percentage) {
        radius *= (1.0 + percentage / 100.0);
        System.out.println("Resizing the circle by " + percentage + "%");
    }}
```

Which of the following statements is true about the code? "You can choose many options."

A) The **Circle** class violates encapsulation principles.

B) The **Circle** class demonstrates method overloading.

C) The **Resizable** interface is an example of multiple inheritance.

D) The **draw** method in the **Circle** class is not accessible from outside the class.

**Ans:**

**Q5** : What is the difference between method overloading and method overriding in Java?

**2 small paragraphs**

What is the purpose of the **final** keyword in Java when applied to a class? "You can choose many options."

A) It indicates that the class cannot be extended, and no subclasses can be created.

B) It indicates that the class cannot have any instance variables.

C) It indicates that the class cannot have any methods.

D) It indicates that the class cannot be instantiated, and no objects of that class can be

**Ans:**

**Q6**: What is the purpose of the **static** keyword in Java? "You can choose many options."

A) It is used to declare a variable that is accessible only within the current method.

B) It is used to create an instance of a class.

C) It is used to declare a variable that is shared among all instances of the class.

D) It is used to indicate that a class cannot be instantiated.

**Ans:**