

1. What is the output of the following program?

```
class A {  
    public void print() {  
        System.out.println("A");  
    }  
}  
  
class B extends A {  
    public void print() {  
        System.out.println("B");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        A obj = new B();  
        obj.print();  
    }  
}
```

- a. A
- b. B
- c. Error
- d. None

2. In the following code, identify the constructor used.

```
class A {  
    public A() {  
        System.out.println("A");  
    }  
    public void B(){  
        System.out.println("B");  
    }  
}  
  
class B extends A {  
    public void C() {  
        System.out.println("Child");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        B obj = new B();  
        System.out.print("The constructor is: ");  
        obj.C();  
    }  
}
```

- a. A
- b. B
- c. C
- d. No constructor is used

3. Select the correct output of the following program:

```
public class MyClass {  
    public static void main(String[] args) {  
        Parent obj1 = new Parent();  
        Parent obj2 = new Child();  
        obj1.printMessage();  
        obj2.printMessage();  
    }  
}  
  
class Parent {  
    public void printMessage() {  
        System.out.println("Parent");  
    }  
}  
  
class Child extends Parent {  
    public void printMessage() {  
        System.out.println("Child");  
        super.printMessage();  
    }  
}
```

- a. Parent  
Child  
Parent
- b. Parent  
Parent  
Child
- c. Parent  
Child  
Child
- d. Error

3. When does method overloading is determined?

- a. At run time
- b. At compile time
- c. At coding time
- d. At execution time

4. When does not method overloading occur?

- a. By changing number of arguments
- b. By changing the data type
- c. By changing the sequence of the data type
- d. By changing the return type

5. What is the output of the following program?

```
class A {  
    public A() {  
        System.out.println("A");  
    }  
    public A(int x){  
        System.out.println("Hello");  
    }  
}  
  
class B extends A {  
    public B() {  
        super(1);  
        System.out.println("B");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        B obj = new B();  
    }  
}
```

- a. A  
B
- b. Hello  
B
- c. B
- d. Error

6. Select the correct output of the following code.

```
class A {  
    public A() {  
        System.out.println("A");  
    }  
    public A(int x){  
        System.out.println("Hello");  
    }  
}  
  
class B extends A {  
    public B() {  
        System.out.println("B");  
        super(1);  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        B obj = new B();  
    }  
}
```

- a. B  
Hello
- b. B  
A
- c. B
- d. Error

7. Predict the output of the following programs:

a.

```
interface Shape {
    void draw();
}

class Circle implements Shape {
    public void draw() {
        System.out.println("Drawing a circle.");
    }
}

class Rectangle implements Shape {
    public void draw() {
        System.out.println("Drawing a rectangle.");
    }
}

public class Main {
    public static void main(String[] args) {
        Shape circle = new Circle();
        circle.draw();

        Shape rectangle = new Rectangle();
        rectangle.draw();
    }
}
```

b.

```
abstract class Vehicle {
    private String brand;

    public Vehicle(String brand) {
        this.brand = brand;
    }

    public abstract void start();

    public void displayBrand() {
        System.out.println("Brand: " + brand);
    }
}

class Car extends Vehicle {
    public Car(String brand) {
        super(brand);
    }

    public void start() {
        System.out.println("Car started.");
    }
}

public class Main {
    public static void main(String[] args) {
        Car vehicle = new Car("Toyota");
        vehicle.start();
        vehicle.displayBrand();
    }
}
```

c.

```
class A {
    public A() {
        System.out.println("A");
    }
}

class B extends A {
    int a;
    public B() {
        super();
        System.out.println(a);
    }
}

public class Main {
    public static void main(String[] args) {
        B obj = new B();
        obj.a=1;
    }
}
```