Variant 2

Part A

1. What is the special character in the end of expression-statement?
2. How many data types intended to represent numbers are there in Java?
3. Can we have an access to the instance field of a class from the static method?
4. How the instances of a primitive type are represented in memory: by value or by reference?
5. Can we overload instance method?
6. Can we override abstract method?

Part B

1. What will be the outcome of the following program?

```
public class ArraysInJava
{
    static final int[] a;

    static
    {
        a = new int[] {1, 2, 3};
    }

    public static void main(String[] args)
    {
        a = new int[5];
    }
}
```

2. What happens when you compile and run the following program?

```
public class ArraysInJava
{
    public static void main(String[] args)
    {
         int[][] a = {{1,2,}, {3,4}};

         int[] b = (int[]) a[1];

         Object o1 = a;

         int[][] a2 = (int[][]) o1;

         int[] b2 = (int[]) o1;

         System.out.println(b[1]);
    }
}
```

3. What will be the output of the below program?

```
class A
{
    static String s = "AAA";

    class B
    {
        String s = "BBB";

        void methodB()
        {
            System.out.println(s);
        }
    }
}
```

```
public class MainClass
{
    public static void main(String[] args)
    {
        A a = new A();

        System.out.println(a.s);

        A.B b = a.new B();

        System.out.println(b.s);

        b.methodB();
    }
}
```

4. What will be the output of the following program?

```
class P
{
    String s = "PPP";

    {
        System.out.println(s);
    }

    String methodP()
    {
        class Q
        {
            String s = P.this.s+"QQQ";

            {
                System.out.println(s);
            }
        }

        return new Q().s+s;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        P p = new P();

        System.out.println(p.methodP());
    }
}
```

5. What will be the outcome of the following program?

```
class X
{
    int x = 111;

    static class Y extends X
    {
        int y = x + 222;
    }

    class Z extends X.Y
    {
        int z = y + 333;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        X.Z z = new X().new Z();
```

```
        System.out.println(z.x);

        System.out.println(z.y);

        System.out.println(z.z);
    }
}
```

6. What will be the output of the following program?

```
abstract class A
{
    abstract void myMethod(Number N);
}

interface B
{
    abstract void myMethod(Object O);
}

class C extends A implements B
{
    void myMethod(Number N)
    {
        System.out.println("Number");
    }

    public void myMethod(Object O)
    {
        System.out.println("Object");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        A a = new C();

        a.myMethod(new Integer(121));

        B b = new C();

        b.myMethod(new Integer(121));

        C c = new C();

        c.myMethod(new Integer(121));
    }
}
```

7. What will be the output of the following program?

```
class ClassOne
{
    int methodOne(int i, int j)
    {
        return i++ + ++j - ++i - j++;
    }
}

abstract class ClassTwo extends ClassOne
{
    abstract int methodOne(int i, int j, int k);

    @Override
    int methodOne(int i, int j)
    {
        return methodOne(i, j, i+j);
    }
}
```

```
class ClassThree extends ClassTwo
{
    @Override
    int methodOne(int i, int j, int k)
    {
        return --i - j-- + ++k - i++ + ++j - k--;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        ClassOne one = new ClassOne();

        ClassThree three = new ClassThree();

        System.out.println(three.methodOne(one.methodOne(10101, 20202), one.methodOne(20202,
10101)));
    }
}
```

8. What will be the output of the following program?

```
class X
{
    int method(int i)
    {
        return i *= i;
    }
}

class Y extends X
{
    double method(double d)
    {
        return d /= d;
    }
}

class Z extends Y
{
    float method(float f)
    {
        return f += f;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Z z = new Z();

        System.out.println(z.method(21.12));
    }
}
```