Variant 4

Part A

1. How many ways of data organization have we discussed?
2. What is the common name for an exact values inside of the source code?
3. Of these given keywords: public, static, final, private, - which are the access modifiers?
4. How many methods of the same name can we have in the program?
5. Can we override static method?
6. Can a class inherit members from three other classes?

Part B

1. What will be the output of this program?

```
public class ArraysInJava
{
    static Double[] methodOne(Double[] D)
    {
        D[1] = 36.25;

        return methodTwo(D);
    }

    static Double[] methodTwo(Double[] D)
    {
        D[1] = 62.36;

        return methodThree(D);
    }

    static Double[] methodThree(Double[] D)
    {
        D[1] = 93.58;

        return D;
    }

    public static void main(String[] args)
    {
        Double[] D = {10.55, 25.36, 58.29, 74.32, 32.21};

        D = methodOne(D);

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

2. Does below program print “SUCCESS” on the console when you run it?

```
class A
{
    {
        new B();
    }

    static class B
    {
        {
            new A().new C();
        }
    }

    class C
    {
        {
            System.out.println("SUCCESS");
        }
    }
}

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

3. Is the below code written correctly?

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

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

    static class B
    {
        void methodB()
        {
            methodA();
        }
    }
}
```

4. Is the below program written correctly? If yes, what will be the output?

```
class One
{
    {
        System.out.println("ONE");
    }

    class Two
    {
        {
            System.out.println("TWO");
        }
    }

    static
    {
        System.out.println("THREE");
    }

    static class Three
    {
        {
            System.out.println("FOUR");
        }

        static
        {
            System.out.println("FIVE");
        }
    }
}

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

        One.Two two = one.new Two();
        One.Three three = new One.Three();
    }
}
```

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

```
class OuterClass
{
    static class InnerClassOne
    {
        int i = 1221;

        int getI()
        {
            return i = i++ - ++i;
        }
    }

    static class InnerClassTwo extends InnerClassOne
    {
        @Override
        int getI()
        {
            return i = i-- + --i;
        }
    }
}

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

        System.out.println(one.getI());

        one = new OuterClass.InnerClassTwo();

        System.out.println(one.getI());
    }
}
```

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

```
interface A
{
    int methodA();
}

interface B
{
    int methodB();
}

interface C
{
    int methodC();
}

class D implements A, B, C
{
    int i = 999+111;

    public int methodA()
    {
        i =+ i / i;

        return i;
    }
    public int methodB()
    {
        i =- i * i;

        return i;
    }

    public int methodC()
    {
        i = ++i - --i;

        return i;
    }
}

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

        System.out.println(d.i);

        System.out.println(d.methodA());

        System.out.println(d.methodB());

        System.out.println(d.methodC());
    }
}
```

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

```
abstract class A
{
    int i = 111, j = 222;

    abstract void methodOne();

    abstract void methodTwo();
}

abstract class B extends A
{
    @Override
    void methodOne()
    {
        System.out.println(i);

        System.out.println(j);

        i = ++i;

        j = --j;
    }
}

class C extends B
{
    @Override
    void methodTwo()
    {
        System.out.println(i);

        System.out.println(j);

        i = i++;

        j = j--;
    }
}

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

        c.methodOne();

        c.methodTwo();

        System.out.println(c.i);

        System.out.println(c.j);
    }
}
```

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

```
class X
{
    void calculate(int a, int b)
    {
        System.out.println("Class X");
    }
}

class Y extends X
{
    @Override
    void calculate(int a, int b)
    {
        System.out.println("Class Y");
    }
}

class Z extends Y
{
    @Override
    void calculate(int a, int b)
    {
        System.out.println("Class Z");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        X x = new Y();

        x.calculate(10, 20);

        Y y = (Y) x;

        y.calculate(50, 100);

        Z z = (Z) y;

        z.calculate(100, 200);
    }
}
```