Variant 1

Part A

1. How many branches can we describe using if-statement at most?
2. How many builtin data types are there in Java?
3. What is the modifier to tell that the member shoud be available only from inside of the class?
4. How the instances of an array type are represented in memory: by value or by reference?
5. Can we have an access to the static method from the instance method?
6. Can we override field?

Part B

1. What will be the output of this program?

```
public class ArraysInJava
{
    public static void main(String[] args)
    {
        int[] a = new int[3];

        a[1] = 50;

        Object o = a;

        int[] b = (int[])o;

        b[1] = 100;

        System.out.println(a[1]);

        ((int[])o)[1] = 500;

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

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

```
public class ArraysInJava
{
    public static void main(String[] args)
    {
        int[] a = {1, 2, 3, 4, 5, 8};

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

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

```
class X
{
    {
        System.out.println(1);
    }

    static
    {
        System.out.println(2);
    }

    public X()
    {
        new Y();
    }

    static class Y
    {
        {
            System.out.println(3);
        }

        static
        {
            System.out.println(4);
        }
    }
}

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

        X.Y y = new X.Y();
    }
}
```

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

```
class ABC
{
    int i = 10101;

    {
        i--;
    }

    public ABC()
    {
        --i;
    }

    class XYZ
    {
        int i = this.i;

        {
            i++;
        }

        public XYZ()
        {
            ++i;
        }
    }
}

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

        System.out.println(abc.i);

        ABC.XYZ xyz = abc.new XYZ();

        System.out.println(xyz.i);

        ABC.XYZ xyz1 = new ABC().new XYZ();

        System.out.println(xyz1.i);
    }
}
```

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

```
class A
{
    int i = 1;

    class B
    {
        int i = 2;

        class C extends A
        {
            void methodC()
            {
                System.out.println(i);
            }
        }
    }
}

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

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

```
interface A
{
    String A = "AAA";

    String methodA();
}

interface B
{
    String B = "BBB";

    String methodB();
}

class C implements A, B
{
    public String methodA()
    {
        return A+B;
    }

    public String methodB()
    {
        return B+A;
    }
}

class D extends C implements A, B
{
    String D = "DDD";

    public String methodA()
    {
        return D+methodB();
    }
}

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

        System.out.println(c.methodA());
        System.out.println(c.methodB());

        c = new D();

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

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

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

```
abstract class A
{
    abstract int firstMethod(int i);

    abstract int secondMethod(int i);

    int thirdMethod(int i)
    {
        return secondMethod(++i);
    }
}

abstract class B extends A
{
    @Override
    int secondMethod(int i)
    {
        return firstMethod(++i);
    }
}

class C extends B
{
    @Override
    int firstMethod(int i)
    {
        return ++i;
    }
}

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

        System.out.println(c.thirdMethod(121121));
    }
}
```

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

```
class A
{
    static void methodOne()
    {
        System.out.println("AAA");
    }
}

class B extends A
{
    static void methodOne()
    {
        System.out.println("BBB");
    }
}

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

        a.methodOne();
    }
}
```