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

```
class A {

}

class B extends A {

}

class C extends B {

}

public class MainClass {

    static void overloadedMethod(A a)
    {
        System.out.println("ONE");
    }

    static void overloadedMethod(B b)
    {
        System.out.println("TWO");
    }

    static void overloadedMethod(Object obj)
    {
        System.out.println("THREE");
    }

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

        overloadedMethod(c);
    }
}
```

Answer 4: TWO

Question 5: In the below class, is ‘method’ overloaded or duplicated?

```
public class MainClass {

      void method(int ... a)
      {
           System.out.println(1);
```

```
    }

    void method(int[] a)
    {
        System.out.println(2);
    }
}
```

Answer 5: Duplicated. Because var args (int ... a) is not but we are taking many arguments at the same time and in (int[] a) also we are taking many arguments at same time . So here, (int ... a) and (int[] a) are the same.

Question 6: What will be the outcome of the below program?

```
public class MainClass {

    double overloadedMethod(double d)
    {
        return d *= d;
    }

    int overloadedMethod(int i)
    {
        return overloadedMethod(i *= i);
    }

    float overloadedMethod(float f)
    {
        return overloadedMethod(f *= f);
    }

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

         System.out.println(main.overloadedMethod(100));
    }
}
```

Answer 6: It will throw java.lang.StackOverflowError at run time.
Because, overloadedMethod(int) keeps
calling itself.

Question 7. What is the output of the following program?

```
class Test {
   void myMethod()
```

```
    {
        System.out.println("Gaurav");
    }
}

public class Derived extends Test {

    void myMethod()
    {
        System.out.println("GFG");
    }

    public static void main(String[] args)
    {
        Derived object = new Test();
        object.myMethod();
    }
}
```

Answer 7: Compilation error. Because we cannot assign a parent class reference object to the child class.

Question 8. What is the output of the following program?

```
interface GFG {

    void myMethod();
    void getInfo();
}

abstract class Gaurav implements GFG {

    void getData()
    {
        System.out.println("GFG");
    }
}

public class Test extends Gaurav {

    public void myMethod()
    {
        System.out.println("Gaurav");
    }
    public void getInfo()
    {
        System.out.println("Geek");
    }

    public static void main(String[] args)
    {
        Gaurav obj = new Test();
        obj.getInfo();
    }
}
```

Answer 8: Geek. Java Abstract class can implement interfaces without even providing the implementation
of interface methods. Here obj is the Gaurav class object so the method getInfo is called of the Gaurav class.

Question 9. What is the output of the following program?

```
// Override of static method
class Parent {

    // static method
    static void show()
    {
        System.out.println("Parent");
    }
}

// Parent inherit in Child class
class Child extends Parent {

    // override show() of Parent
    void show()
    {
        System.out.println("Child");
    }
}

class GFG {
    public static void main(String[] args)
    {
        Parent p = new Parent();
        // calling Parent's show()
        p.show();
        // cannot override Parent's show()
    }
}
```

Answer 9: Instance method ‘void show ’ in ‘child’ cannot override a static parent class method.

Question 10:

```
public class Myclass {
    private int x = 10;
    static int m1() {
        int y = x;
        return y;
    }
    public static void main(String[] args) {
         m1();
    }
}
```

Answer 10: Non static member “X” cannot be referenced from static context.

Question 11: What is the output of the program?

```
public class StaticDemo {

static String n1= examName("O");{
     n1=examName("A");
  }
static{
      n1=examName("C");
}
public static void main(String[] args) {
           StaticDemo sd = new StaticDemo();

}

public static String examName(String s){
          System.out.println(s);
        return s;
   }
}
```

Answer 11: O,C,A will be the sequence of the output.