1. A class called **circle** is designed as shown in the following class diagram. It contains:

* Two private instance variables: radius (of the type double) and color (of the type String), with default value of 1.0 and "red", respectively.
* Two *overloaded* constructors - a *default* constructor with no argument, and a constructor which takes a double argument for radius.
* Two public methods: getRadius() and getArea(), which return the radius and area of this instance, respectively.

**/\***

**\* The Circle class models a circle with a radius and color.**

**\*/**

**public class Circle { // Save as "Circle.java"**

**// private instance variable, not accessible from outside this class**

**private double radius;**

**private String color;**

**// The default constructor with no argument.**

**// It sets the radius and color to their default value.**

**public Circle() {**

**radius = 1.0;**

**color = "red";**

**}**

**// 2nd constructor with given radius, but color default**

**public Circle(double r) {**

**radius = r;**

**color = "red";**

**}**

**// A public method for retrieving the radius**

**public double getRadius() {**

**return radius;**

**}**

**// A public method for computing the area of circle**

**public double getArea() {**

**return radius\*radius\*Math.PI;**

**}**

**}**

2. Identify the Error

class X

{

    //Class X Members

}

class Y

{

    //Class Y Members

}

class Z extends X, Y

{

    //Class Z Members

}

Ans - In Java, a class can not extend more than one class. Class Z is extending two classes – Class X and Class Y. It is a compile time error in java.

3. **What will be the output of this program?**

class A

{

    int i = 10;

}

class B extends A

{

    int i = 20;

}

public class MainClass

{

    public static void main(String[] args)

    {

        A a = new B();

        System.out.println(a.i);

    }

}

**ANS: 10**

**4. What will be the output of this program?**

class A

{

    {

        System.out.println(1);

    }

}

class B extends A

{

    {

        System.out.println(2);

    }

}

class C extends B

{

    {

        System.out.println(3);

    }

}

public class MainClass

{

    public static void main(String[] args)

    {

        C c = new C();

    }

}

O/P: 1 2 3

**5. What will be the output of this program?**

class A

{

    String s = "Class A";

}

class B extends A

{

    String s = "Class B";

    {

        System.out.println(super.s);

    }

}

class C extends B

{

    String s = "Class C";

    {

        System.out.println(super.s);

    }

}

public class MainClass

{

    public static void main(String[] args)

    {

        C c = new C();

        System.out.println(c.s);

    }

}

**O/P:** Class A  
Class B  
Class C

6. **What will be the output of this program?**

class A

{

    static

    {

        System.out.println("THIRD");

    }

}

class B extends A

{

    static

    {

        System.out.println("SECOND");

    }

}

class C extends B

{

    static

    {

        System.out.println("FIRST");

    }

}

public class MainClass

{

    public static void main(String[] args)

    {

        C c = new C();

    }

}

**O/P:** THIRD  
SECOND  
FIRST

7. **What will be the output of this program?**

class A

{

    public A()

    {

        System.out.println("Class A Constructor");

    }

}

class B extends A

{

    public B()

    {

        System.out.println("Class B Constructor");

    }

}

class C extends B

{

    public C()

    {

        System.out.println("Class C Constructor");

    }

}

public class MainClass

{

    public static void main(String[] args)

    {

        C c = new C();

    }

}

**O/p:**

Class A Constructor  
Class B Constructor  
Class C Constructor

8. **Private members of a class are inherited to sub class. True or false? - False**

9. class X

{

    static void staticMethod()

    {

        System.out.println("Class X");

    }

}

class Y extends X

{

    static void staticMethod()

    {

        System.out.println("Class Y");

    }

}

public class MainClass

{

    public static void main(String[] args)

    {

        Y.staticMethod();

    }

}

**O/P: Class Y**

**10.** **Below code is showing compile time error. Can you suggest the corrections?**

class X

{

    public X(int i)

    {

        System.out.println(1);

    }

}

class Y extends X

{

    public Y()

    {

**Super(10); //Correction – explicitly constructor calling -----Ans**

System.out.println(2);

    }

}

**11. Can we use this() and super() in a method? –NO**

**Eg.** class SuperClass

{

    public SuperClass()

    {

        System.out.println("Super Class Constructor");

    }

}

class SubClass extends SuperClass

{

    public SubClass()

    {

        System.out.println("Sub Class Constructor");

    }

    void method()

    {

        this();     //Compile time error

        super();    //Compile time error

    }

}

**12. Can you create an object without using new operator in Java?**

Yes, We can create an object without using new operator. There are some other ways to create objects other than using new operator. But, 95% of object creation in java is done through new operator only.

13. **How do you restrict a member of a class from inheriting to it’s sub classes.?**

By declaring that member as a private. Because, private members are not inherited to sub classes.

14. **Can a class extend itself.?**

No, A class can not extend itself.

15. **What happens if both, super class and sub class, have a field with same name.?**

Super class field will be hidden in the sub class. You can access hidden super class field in sub class using super keyword.

**16. Are static members inherited to sub classes?**

Yes, Static members are also inherited to sub classes.