**Date:15-sep-2020**

**Review Questions on Inheritance**

##### ***Part-A:***

##### ***Answer the following in two lines(Minimum)***

##### Why do we need inheritance?

1. Assume we have two classes as Employee and Customer with similar properties as name, age, mobileno, adharnumber, city and method as void printDetails(). Is it right to apply inheritance here?

##### What is the super class for all the classes?

##### What type of inheritance does java support?

##### Can a class extend more than one class?

##### Can I access the sub class methods using a super class object?

##### Point out the difference between Inheritance and Association/ composition?

1. Can we use super keyword in static method of a sub class for calling parent class method?
2. Why can’t we use a super keyword in static method of a sub class for calling parent class method?

### Point out the difference between this() and super()?

### Can I use this and super keyword in a method?

### Can you inherit a constructor?

### What is method overriding? Can I override a private method?

#### Can subclass inherit static members?

### What happens if the parent and the child class have a field with same identifier?

### How do you restrict a member of a class from inheriting by its sub classes?

### Can we reduce the visibility of the inherited or overridden method?

***Part-B:***

***What will be the output of the following program? If it compiles successfully****,* ***write the output otherwise point out the error and include suitable statements to correct it.***

1.

|  |
| --- |
| class X  {      static void display()      {          System.out.println("Class X");      }  }  class Y extends X  {      static void display()      {          System.out.println("Class Y");      }  }   public class Demo  {      public static void main(String[] args)      {          Y.display();      }  } |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.

class X

{

    static void display()

    {

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

    }

}

 class Y extends X

{

    static void display()

    {

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

    }

}

 public class demo

{

    public static void main(String[] args)

    {

        X obj=new Y;

obj.display();

    }

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.

class X

{

    public X(int i)

    {

        System.out.println(“X”);

    }

}

class Y extends X

{

    public Y()

    {

        System.out.println(“Y”);

    }

}

public class Demo

{

    public static void main(String[] args)

    {

        Y obj = new Y();

     }}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.

class A

{

    int i = 10;

}

class B extends A

{

    int i = 20;

}

public class Demo

{

    public static void main(String[] args)

    {

        A a = new B();

        System.out.println(a.i);

    }}

5.

class X

{

 public X(int i)

    {

        System.out.println(“X”);

    }

}

class Y extends X

{

   protected int i=10;

public int j=5;

public Y()

    {

        System.out.println(“Y”);

    }

}

public class Demo

{

    public static void main(String[] args)

    {

        X obj = new X();

System.out.println(obj.i);

System.out.println(obj.j);

     }}