## OOAD Assignment 1 and 2

## 1) In a class, one method has two overloaded forms. One form is defined as static and another form is defined as non-static. Is that method properly overloaded?

A) YES, but we need to initialize the class to call both static and non-static methods.

CODE:

public class Main {

    public static void form(int i) {

        System.out.println("Given parameter is: " + i);

    }

     public void  form(double d) {

        System.out.println("Given parameter is: " + d);

    }

    public static void main(String[] args) {

        Main m = new Main();

        m.form(1);

        m.form(1.1);

    }

}

O/P:

Given parameter is: 1

Given parameter is: 1.1

## 

2) In the below Class X, is ‘method’ properly overloaded?

## class X

## {

## int method(int i, int d)

## {

## return i+d;

## }

## 

## static int method(int i, double d)

## {

## return (int)(i+d);

## }

## 

## double method(double i, int d)

## {

## return i+d;

## }

## 

## static double method(double i, double d)

## {

## return i+d;

## }

## }

A) YES, ‘method’ is properly overloaded. Use of static keyword doesn’t make a difference in method overloading. As long as method signatures are different code will compile.

class X

{

    int method(int i, int d)

    {

        return i+d;

    }

    static int method(int i, double d)

    {

        return (int)(i+d);

    }

    double method(double i, int d)

    {

        return i+d;

    }

    static double method(double i, double d)

    {

        return i+d;

    }

}

public class Main {

    public static void main(String[] args) {

        X x = new X();

        System.out.println((x.method(1, 1)));

        System.out.println(x.method(1, 1.1));

        System.out.println(x.method(1.1, 1));

        System.out.println(x.method(1.1, 1.1));

    }

}

O/P:

2

2

2.1

2.2