**DEPARTMENT OF COMPUTER SCIENCE & APPLICATION**

**MAHARSHI DAYANAND UNIVERSITY**

**ROHTAK**

**Lab-Assignment**

**Subject:- Advance Data Structure**

**By:- Deepak**

**Roll No. :- 21149**

**Class:- MCA-1st Sem.**

**Q.1) Write a Program in which you will declare a default method in the base class and use that method in child class in the same package.**

**Package hy1;**

**Class baseclass {**

**Void print(){**

**System.out.println(“This is base class”);**

**}}**

**Class class1 extends baseclass{**

**Public static void main(String Args[]){**

**Baseclass obj=new baseclass();**

**Obj.print();**

**}**

**}**

**OUTPUT**

**Q.2) Write a Program in which you will declare and use the default method in the same class.**

**Class program2 {**

**Void print(){**

**System.out.println(“use of default method in same class”);**

**}**

**Public static void main(String Args[]){**

**Program2 obj=new program2();**

**Obj.print();**

**}**

**}**

**OUTPUT**

**Q.3) Write a Program in which you will declare and use private methods in the same class in same package.**

**Package hy3;**

**Class program3 {**

**Private void print(){**

**System.out.println(“use of private method in same class of same package”);**

**}**

**Public static void main(String args[]){**

**Program3 obj=new program3();**

**Obj.print();**

**}**

**}**

**OUTPUT**

**Q.4) Write a Program in which you will declare a protected method in the base class and use that method in the child class in the same package.**

**Package hy4;**

**Class base{**

**Protected void print(){**

**System.out.println(“use of protected method of base class in child class of same package”);**

**}**

**}**

**Class chlid extends base{**

**Void display(){**

**Base obj=new base();**

**Obj.print();**

**}}**

**Public class program4{**

**Public void display2(){**

**Chlid obj2=new chlid();**

**Obj2.display();**

**}**

**Public static void main(String args[]){**

**Program4 sc=new program4();**

**Sc.display2();**

**}}**

**OUTPUT**

**Q.5) Write a Program in which you will declare and use the protected method in the same class.**

**Class program5 {**

**Protected void print(){**

**System.out.println(“Use of protected method in same class”);**

**}**

**Public static void main(String args[]){**

**Program5 obj=new program5();**

**Obj.print();**

**}**

**}**

**OUTPUT**

**Q.6) Write a Program in which you will declare a protected method in a class and use that method in a non-child class of the same package.**

**Package hy6;**

**Class demo {**

**Protected void print(){**

**System.out.print(“use of protected method in non-chlid class of same package”);**

**}**

**}**

**Class program6{**

**Public static void main(String args[]){**

**Demo obj=new demo();**

**Obj.print();**

**}**

**}**

**OUTPUT**

**Q.7) Write a Program in which you will declare a protected method in the base class of a different package and use that method in the child class of a different package.**

**Package hyy7;**

**Public class seven{**

**Protected static void display(){**

**System.out.println(“This method belongs to package hyseven class myClass”);**

**}**

**}**

**Package hhy7;**

**Import hyy7.seven;**

**Public class program7 extends seven {**

**Public static void main(String[] args) {**

**Display();**

**}**

**}**

**OUTPUT**

**Q.8 Write a Java program to find the factorial of a number.**

**Import java.util.Scanner;**

**Class Factorial{**

**Public static void main(String args[]){**

**Scanner sc=new Scanner (System.in);**

**System.out.println(“Enter the no.= “);**

**Int a=sc.nextInt();**

**Int i=a,fact=1;**

**For( i=1;i<=a;i++){**

**Fact=fact\*I;**

**}**

**System.out.println(“Factorial is = “+fact);**

**}}**

**OUTPUT**

**Q.9 Write a java program that initialization earning of an employee. The program shoul**

**Calculate the income tax to be paid by the employee as per the criteria given below:**

**Slab rate IT rate**

**Up to Rs. 80,000 Nil**

**Up to Rs. 90,000 10% on additional amount**

**Up to Rs. 1, 30,000 20% on additional amount**

**Above Rs. 1, 30,000 30% on additional amount**

**Hint: - Run: - java calculates 1, 20,000**

**Result: - income tax is…………………………….**

**Import java.util.Scanner;**

**Class incometax{**

**Public static void main(String args[]){**

**Double a,tax = 0;**

**Scanner sc=new Scanner(System.in);**

**System.out.print(“enter income: “);**

**A=sc.nextDouble();**

**If(a<=80000){**

**Tax=0;**

**}**

**Else if(a<=90000){**

**Tax=(a-80000)\*0.1;**

**}**

**Else if(a<=130000){**

**Tax=(a-90000)\*0.2 + (10000)\*0.1;**

**}**

**Else if(a>=130000){**

**Tax=(a-130000)\*0.3 + (40000)\*0.2 + (10000)\*0.1;;**

**}**

**System.out.println(“Income tax amount is “+tax);**

**}**

**}**

**OUTPUT**

**Q.10 Write a program to print the following pattern given n as argument:-**

**A B C D E F F E D C B A**

**A B C D E E D C B A**

**A B C D D C B A**

**A B C C B A**

**A B B A**

**A A**

**A A**

**A B B A**

**A B C C B A**

**A B C D D C B A**

**A B C D E E D C B A**

**A B C D E F F E D C B A**

**Import java.util.Scanner;**

**Class program\_10{**

**Public static void main(String args[])**

**{**

**Scanner in = new Scanner(System.in);**

**String s = “ABCDEFGHIJKLMNOPQRSTUVWXYZ”;**

**Int g, I, j, n;**

**System.out.print(“Enter the number of alphabets to be used: “);**

**N = in.nextInt();**

**For(i=1; i<=n; i++)**

**{**

**For(j=1; j<=n-(i-1); j++)**

**{**

**System.out.print(s.charAt(j-1)+” “);**

**}**

**For(j = n-i+1; j>0; j--)**

**{**

**System.out.print(s.charAt(j-1)+” “);**

**}**

**System.out.println();**

**}**

**For(i=n; i>0; i--)**

**{**

**For(j=1; j<=n-(i-1); j++)**

**{**

**System.out.print(s.charAt(j-1)+” “);**

**}**

**For(j = n-i+1; j>0; j--)**

**{**

**System.out.print(s.charAt(j-1)+” “);**

**}**

**System.out.println();**

**}**

**In.close();**

**}**

**}**

**OUTPUT**

**Q.11 Write a JAVA program that has**

1. **2 classes which initialize a String in its constructor**
2. **A Generic class with 2 type Parameters**
3. **Create a Generic Class reference for the above 2 Class and try to print the message inside the constructor (Use to string method).**

**Class InitializeString{**

**String message;**

**InitializeString(String message){**

**This.message = message;**

**}**

**}**

**Class InitializeString2{**

**String message;**

**InitializeString2(String message){**

**This.message = message;**

**}**

**}**

**Class PrintString<T1 extends InitializeString,T2 extends InitializeString2> {**

**T1 obj;**

**T2 obj2;**

**PrintString(T1 ob, T2 ob2){**

**This.obj = ob;**

**This.obj2 = ob2;**

**}**

**Void print(){**

**System.out.println(“String stored at object of “ + obj.getClass().getName() + “: “ + obj.message);**

**System.out.println(“String stored at object of “ + obj2.getClass().getName() + “: “ + obj2.message);**

**}**

**}**

**Class Program11 {**

**Public static void main(String[] args) {**

**InitializeString a = new InitializeString(“Hi”);**

**InitializeString2 b = new InitializeString2(“Good Morning”);**

**PrintString<InitializeString, InitializeString2> object = new PrintString<InitializeString, InitializeString2>(a,b);**

**Object.print();**

**}**

**}**

**OUTPUT**

**Q.12 Write a Java Program to demonstrate the use of the Container class.**

**Import java.awt.\*;**

**Import javax.swing.\*;**

**Public class ContainerTest extends JFrame { // top-level container**

**JPanel panel; // low-level container**

**JTextField field;**

**JButton btn;**

**Public ContainerTest() {**

**setTitle(“Container Test”);**

**panel = new JPanel();**

**field = new JTextField(20);**

**panel.add(field);**

**btn = new JButton(“Submit”);**

**panel.add(btn);**

**add(panel, BorderLayout.CENTER);**

**setSize(350, 275);**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**setLocationRelativeTo(null);**

**setVisible(true);**

**}**

**Public static void main(String args[]) {**

**New ContainerTest();**

**}**

**}**

**OUTPUT**

**Q.13- Write a Java program to practice using String Buffer class and its methods.**

**Public class Program13 {**

**Public static void main(String[] args) {**

**StringBuffer message = new StringBuffer(“Hello! “);**

**System.out.println(“StringBuffer: “ + message);**

**Message.append(“What is your Name?”);**

**System.out.println(“StringBuffer after append: “ + message);**

**System.out.println(“Length of StringBuffer: “ + message.length());**

**System.out.println(“Capacity of StringBuffer: “ + message.capacity());**

**System.out.println(“Character at index 5: “ + message.charAt(5));**

**System.out.println(“StringBuffer after deletion: “ + message.delete(0, 5));**

**System.out.println(“Inserting hi at index 0: “ + message.insert(0,”hi”));**

**System.out.println(“Reverse StringBuffer: “ + message.reverse());**

**}**

**}**

**OUTPUT**

**Q.14 – Write a Java Program to implement the Vector class and its methods.**

**Import java.util.Vector;**

**Public class program14 {**

**Public static void main(String[] args) {**

**Vector<Integer> nums = new Vector<>();**

**Nums.add(11);**

**Nums.add(13);**

**Nums.add(20);**

**Nums.add(9);**

**System.out.println(“Values stored in Vector: “);**

**For(int I : nums){**

**System.out.println(i);**

**}**

**Nums.add(1, 100);**

**System.out.println(“After adding 100 to index 1: “);**

**For(int I : nums){**

**System.out.println(i);**

**}**

**System.out.println(“Capacity of Vector: “ + nums.capacity());**

**Vector nums2 = (Vector) nums.clone();**

**System.out.println(“Num 10 constains the vector: “ + nums.contains(10));**

**Nums.clear();**

**}**

**}**

**OUTPUT**

**Q.15- Write a Java Program to implement Wrapper classes and their methods.**

**Public class program15 {**

**Public static void main(String[] args) {**

**Integer I = Integer.valueOf(“10”);**

**System.out.println(“Integer value: “ + I);**

**Double D = Double.valueOf(“10.0”);**

**System.out.println(“Double value: “ + D);**

**Boolean B = Boolean.valueOf(“true”);**

**System.out.println(“Boolean value: “ + B);**

**/\* valueOf(String s, int radix) method -> to create a Wrapper object for the given String with specified radix \*/**

**Integer I1 = Integer.valueOf(“1111”, 2);**

**System.out.println(“Integer value 2: “ + I1);**

**Integer I2 = Integer.valueOf(“1111”, 4);**

**System.out.println(“Integer value 3: “ + I2);**

**/\* xxxValue() method -> to get the primitive for the given Wrapper Object \*/**

**Integer I4 = new Integer(130);**

**System.out.println(“ByteValue of Integer: “ + I4.byteValue());**

**System.out.println(“ShortValue of Integer: “ + I4.shortValue());**

**System.out.println(“IntValue of Integer: “ + I4.intValue());**

**System.out.println(“LongValue of Integer: “ + I4.longValue());**

**System.out.println(“FloatValue of Integer: “ + I4.floatValue());**

**System.out.println(“DoubleValue of Integer: “ + I4.doubleValue());**

**/\* parseXxx() method -> to find primitive for the given String object \*/**

**Int I = Integer.parseInt(“10”);**

**Double d = Double.parseDouble(“10.5”);**

**Boolean b = Boolean.parseBoolean(“true”);**

**System.out.println(“String to Integer: “ + i);**

**System.out.println(“String to Double: “ + d);**

**System.out.println(“String to Boolean: “ + b);**

**/\* toString() method -> to convert Wrapper object or primitive to String \*/**

**Integer I5 = new Integer(10);**

**String s = I5.toString();**

**System.out.println(“Integer to String: “ + s);**

**}**

**}**

**OUTPUT**

**Q.16- Write a Java Program to implement inheritance and demonstrate the use of method overriding.**

**Class Animal {**

**Public void displayInfo() {**

**System.out.println(“I am an animal.”);**

**}**

**}**

**Class Dog extends Animal {**

**@Override**

**Public void displayInfo() {**

**System.out.println(“I am a dog.”);**

**}**

**}**

**Class program16{**

**Public static void main(String[] args) {**

**Dog d1 = new Dog();**

**D1.displayInfo();**

**}**

**}**

**OUTPUT**

**Q.17- Write a program to demonstrate the use of implementing extending interfaces.**

**Interface ParentInterface {**

**Void parentMethod();**

**}**

**Interface ChildInterface extends ParentInterface {**

**Void childMethod();**

**}**

**Class ImplementingClass implements ChildInterface {**

**Public void childMethod() {**

**System.out.println(“Child Interface method!!”);**

**}**

**Public void parentMethod() {**

**System.out.println(“Parent Interface mehtod!”);**

**}**

**}**

**Class ExtendingAnInterface {**

**Public static void main(String[] args) {**

**ImplementingClass obj = new ImplementingClass();**

**Obj.childMethod();**

**obj.parentMethod();**

**}**

**}**

**OUTPUT**

**Q.1) Write a Program in which you will declare a default method in the base class and use that method in child class in the same package.**

package hy1;

class baseclass {

    void print(){

        System.out.println("This is base class");

    }}

class class1 extends baseclass{

    public static void main(String Args[]){

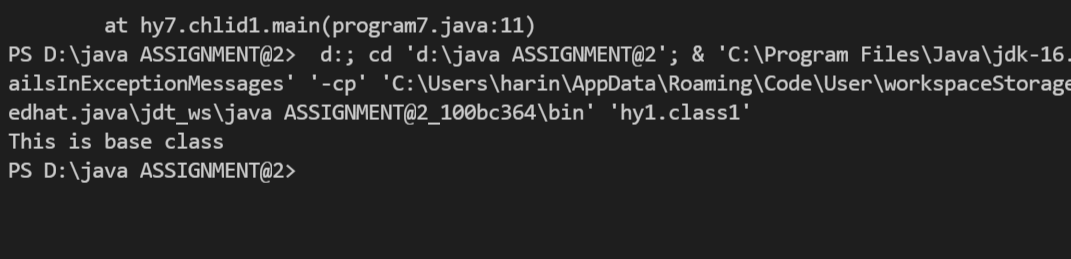
        baseclass obj=new baseclass();

        obj.print();

    }

}

**OUTPUT**

****

**Q.2) Write a Program in which you will declare and use the default method in the same class.**

class program2 {

    void print(){

        System.out.println("use of default method in same class");

    }

    public static void main(String Args[]){

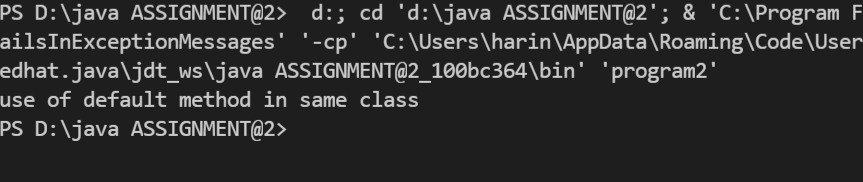
        program2 obj=new program2();

        obj.print();

    }

}

**OUTPUT**

****

**Q.3) Write a Program in which you will declare and use private methods in the same class in same package.**

package hy3;

class program3 {

private void print(){

        System.out.println("use of private method in same class of same package");

    }

    public static void main(String args[]){

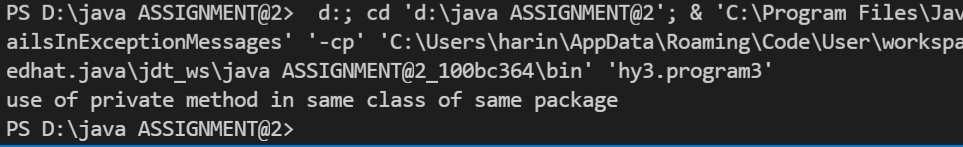
        program3 obj=new program3();

        obj.print();

    }

}

**OUTPUT**



**Q.4) Write a Program in which you will declare a protected method in the base class and use that method in the child class in the same package.**

package hy4;

 class base{

     protected void print(){

         System.out.println("use of protected method of base class  in child class of same package");

     }

}

  class chlid extends base{

    void display(){

    base obj=new base();

    obj.print();

}}

 public class program4{

     public void display2(){

        chlid obj2=new chlid();

        obj2.display();

    }

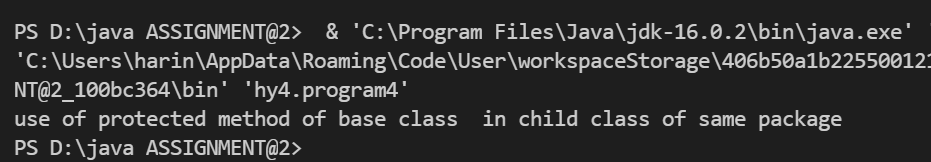
    public static void main(String args[]){

        program4 sc=new program4();

        sc.display2();

    }}

**OUTPUT**



**Q.5) Write a Program in which you will declare and use the protected method in the same class.**

class program5 {

     protected void print(){

         System.out.println("Use of protected method in same class");

     }

    public static void main(String args[]){

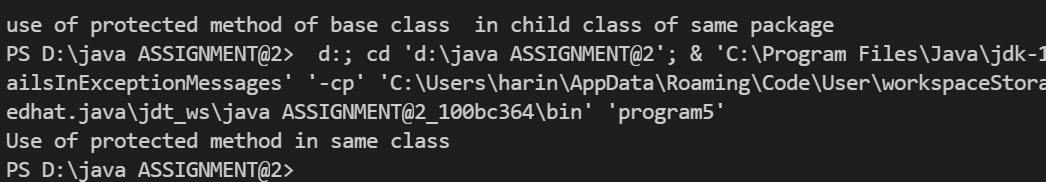
        program5 obj=new program5();

        obj.print();

    }

}

**OUTPUT**



**Q.6) Write a Program in which you will declare a protected method in a class and use that method in a non-child class of the same package.**

package hy6;

 class demo {

     protected void print(){

         System.out.print("use of protected method in non-chlid class of same package");

     }

}

class program6{

    public static void main(String args[]){

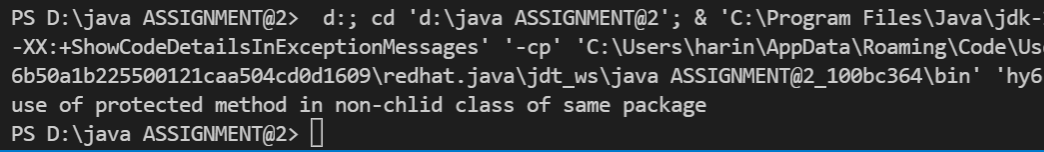
        demo obj=new demo();

        obj.print();

    }

}

**OUTPUT**

****

**Q.7) Write a Program in which you will declare a protected method in the base class of a different package and use that method in the child class of a different package.**

package hyy7;

  public class seven{

    protected static void display(){

        System.out.println("This method belongs to package hyseven class myClass");

    }

}

package hhy7;

import hyy7.seven;

 public class program7 extends seven {

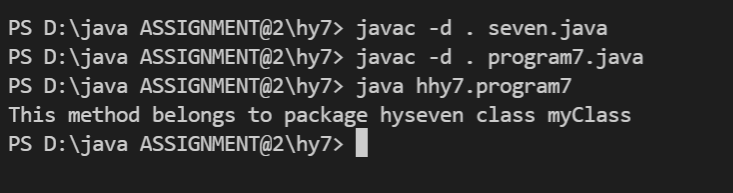
    public static void main(String[] args) {

        display();

    }

}

**OUTPUT**

****

**Q.8 Write a Java program to find the factorial of a number.**

import java.util.Scanner;

class Factorial{

    public static void main(String args[]){

        Scanner sc=new Scanner (System.in);

        System.out.println("Enter the no.= ");

        int a=sc.nextInt();

     int i=a,fact=1;

for( i=1;i<=a;i++){

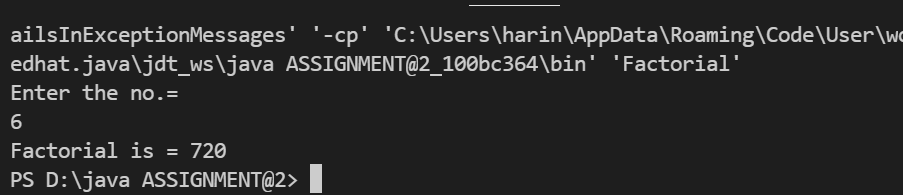
    fact=fact\*i;

}

System.out.println("Factorial is = "+fact);

}}

**OUTPUT**



**Q.9 Write a java program that initialization earning of an employee. The program shoul**

**calculate the income tax to be paid by the employee as per the criteria given below:**

**Slab rate IT rate**

**Up to Rs. 80,000 Nil**

**Up to Rs. 90,000 10% on additional amount**

**Up to Rs. 1, 30,000 20% on additional amount**

**Above Rs. 1, 30,000 30% on additional amount**

**Hint: - Run: - java calculates 1, 20,000**

**Result: - income tax is..................................**

import java.util.Scanner;

class incometax{

    public static void main(String args[]){

        double a,tax = 0;

        Scanner sc=new Scanner(System.in);

        System.out.print("enter income: ");

        a=sc.nextDouble();

        if(a<=80000){

            tax=0;

        }

        else if(a<=90000){

            tax=(a-80000)\*0.1;

        }

        else if(a<=130000){

            tax=(a-90000)\*0.2 + (10000)\*0.1;

        }

        else if(a>=130000){

            tax=(a-130000)\*0.3 + (40000)\*0.2 + (10000)\*0.1;;

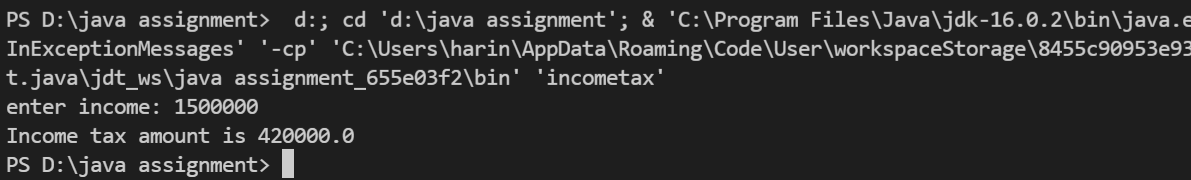
        }

        System.out.println("Income tax amount is "+tax);

    }

}

**OUTPUT**



**Q.10 Write a program to print the following pattern given n as argument:-**

**A B C D E F F E D C B A**

**A B C D E E D C B A**

**A B C D D C B A**

**A B C C B A**

**A B B A**

**A A**

**A A**

**A B B A**

**A B C C B A**

**A B C D D C B A**

**A B C D E E D C B A**

**A B C D E F F E D C B A**

import java.util.Scanner;

 class program\_10{

    public static void main(String args[])

    {

        Scanner in = new Scanner(System.in);

        String s = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

        int g, i, j, n;

        System.out.print("Enter the number of alphabets to be used: ");

        n = in.nextInt();

        for(i=1; i<=n; i++)

        {

            for(j=1; j<=n-(i-1); j++)

            {

                System.out.print(s.charAt(j-1)+" ");

            }

            for(j = n-i+1; j>0; j--)

            {

                System.out.print(s.charAt(j-1)+" ");

            }

            System.out.println();

        }

        for(i=n; i>0; i--)

        {

            for(j=1; j<=n-(i-1); j++)

            {

                System.out.print(s.charAt(j-1)+" ");

            }

            for(j = n-i+1; j>0; j--)

            {

                System.out.print(s.charAt(j-1)+" ");

            }

            System.out.println();

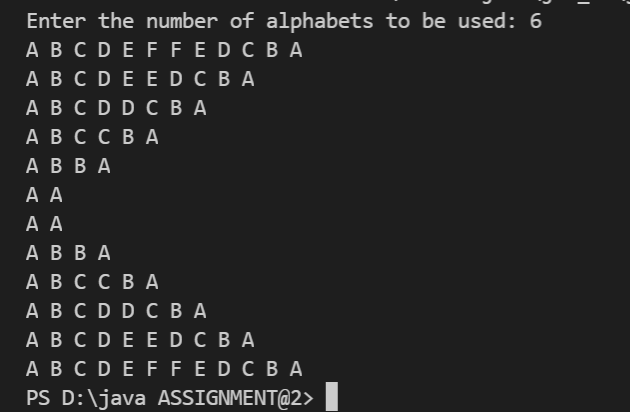
        }

        in.close();

    }

}

**OUTPUT**



**Q.11 Write a JAVA program that has**

**i. 2 classes which initialize a String in its constructor**

**ii. A Generic class with 2 type Parameters**

**iii. Create a Generic Class reference for the above 2 Class and try to print the message inside the constructor (Use to string method).**

class InitializeString{

    String message;

    InitializeString(String message){

        this.message = message;

    }

}

class InitializeString2{

    String message;

    InitializeString2(String message){

        this.message = message;

    }

}

class PrintString<T1 extends InitializeString,T2 extends InitializeString2> {

    T1 obj;

    T2 obj2;

    PrintString(T1 ob, T2 ob2){

        this.obj = ob;

        this.obj2 = ob2;

    }

    void print(){

        System.out.println("String stored at object of " + obj.getClass().getName() + ": " + obj.message);

        System.out.println("String stored at object of " + obj2.getClass().getName() + ": " + obj2.message);

    }

}

class Program11 {

    public static void main(String[] args) {

        InitializeString a = new InitializeString("Hi");

        InitializeString2 b = new InitializeString2("Good Morning");

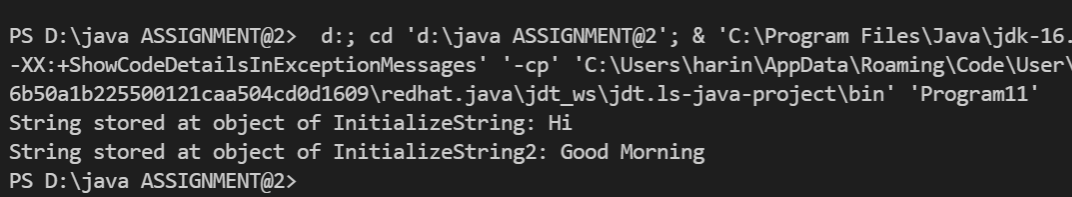
        PrintString<InitializeString, InitializeString2> object = new PrintString<InitializeString, InitializeString2>(a,b);

        object.print();

    }

}

**OUTPUT**

****

**Q.12 Write a Java Program to demonstrate the use of the Container class.**

import java.awt.\*;

import javax.swing.\*;

public class ContainerTest extends JFrame { // top-level container

   JPanel panel; // low-level container

   JTextField field;

   JButton btn;

   public ContainerTest() {

      setTitle("Container Test");

      panel = new JPanel();

      field = new JTextField(20);

      panel.add(field);

      btn = new JButton("Submit");

      panel.add(btn);

      add(panel, BorderLayout.CENTER);

      setSize(350, 275);

      setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

      setLocationRelativeTo(null);

      setVisible(true);

   }

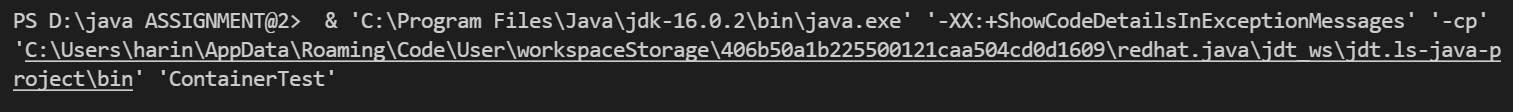
   public static void main(String args[]) {

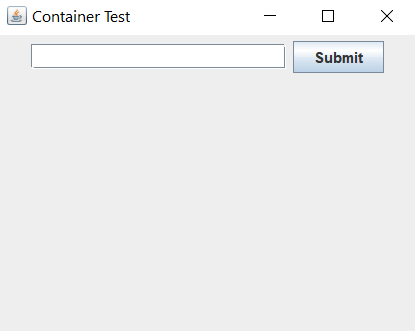
      new ContainerTest();

   }

}

**OUTPUT**

****

****

**Q.13- Write a Java program to practice using String Buffer class and its methods.**

public class Program13 {

    public static void main(String[] args) {

        StringBuffer message = new StringBuffer("Hello! ");

        System.out.println("StringBuffer: " + message);

        message.append("What is your Name?");

        System.out.println("StringBuffer after append: " + message);

        System.out.println("Length of StringBuffer: " + message.length());

        System.out.println("Capacity of StringBuffer: " + message.capacity());

        System.out.println("Character at index 5: " + message.charAt(5));

        System.out.println("StringBuffer after deletion: " + message.delete(0, 5));

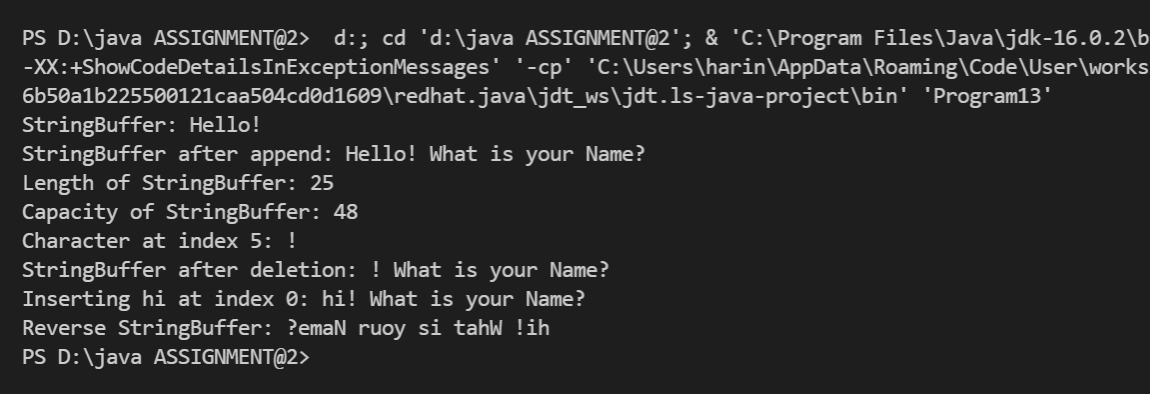
        System.out.println("Inserting hi at index 0: " + message.insert(0,"hi"));

        System.out.println("Reverse StringBuffer: " + message.reverse());

    }

}

**OUTPUT**



**Q.14 - Write a Java Program to implement the Vector class and its methods.**

import java.util.Vector;

public class program14 {

    public static void main(String[] args) {

        Vector<Integer> nums = new Vector<>();

        nums.add(11);

        nums.add(13);

        nums.add(20);

        nums.add(9);

        System.out.println("Values stored in Vector: ");

        for(int i : nums){

            System.out.println(i);

        }

        nums.add(1, 100);

        System.out.println("After adding 100 to index 1: ");

        for(int i : nums){

            System.out.println(i);

        }

        System.out.println("Capacity of Vector: " + nums.capacity());

        Vector nums2 = (Vector) nums.clone();

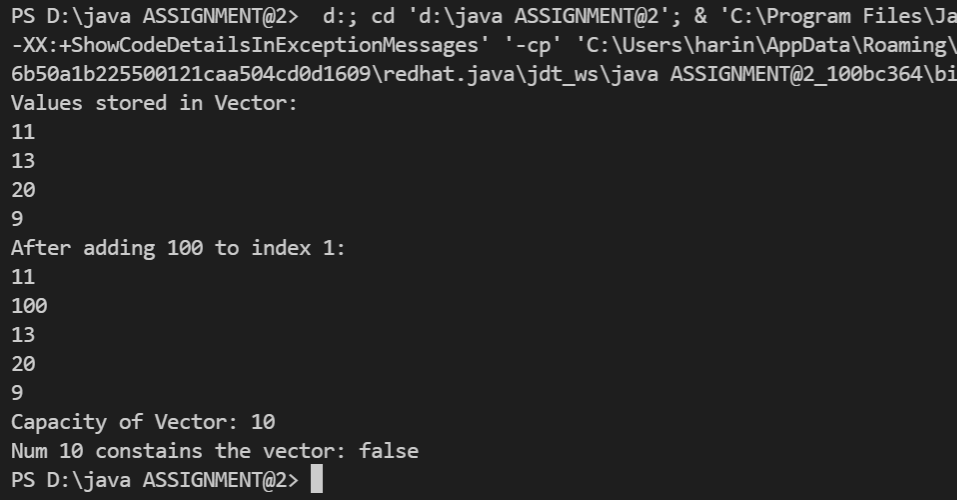
        System.out.println("Num 10 constains the vector: " + nums.contains(10));

        nums.clear();

    }

}

**OUTPUT**



**Q.15- Write a Java Program to implement Wrapper classes and their methods.**

public class program15 {

    public static void main(String[] args) {

        Integer I = Integer.valueOf("10");

        System.out.println("Integer value: " + I);

        Double D = Double.valueOf("10.0");

        System.out.println("Double value: " + D);

        Boolean B = Boolean.valueOf("true");

        System.out.println("Boolean value: " + B);

        /\* valueOf(String s, int radix) method -> to create a Wrapper object for the given String with specified radix \*/

        Integer I1 = Integer.valueOf("1111", 2);

        System.out.println("Integer value 2: " + I1);

        Integer I2 = Integer.valueOf("1111", 4);

        System.out.println("Integer value 3: " + I2);

        /\* xxxValue() method -> to get the primitive for the given Wrapper Object \*/

        Integer I4 = new Integer(130);

        System.out.println("ByteValue of Integer: " + I4.byteValue());

        System.out.println("ShortValue of Integer: " + I4.shortValue());

        System.out.println("IntValue of Integer: " + I4.intValue());

        System.out.println("LongValue of Integer: " + I4.longValue());

        System.out.println("FloatValue of Integer: " + I4.floatValue());

        System.out.println("DoubleValue of Integer: " + I4.doubleValue());

        /\* parseXxx() method -> to find primitive for the given String object \*/

        int i = Integer.parseInt("10");

        double d = Double.parseDouble("10.5");

        boolean b = Boolean.parseBoolean("true");

        System.out.println("String to Integer: " + i);

        System.out.println("String to Double: " + d);

        System.out.println("String to Boolean: " + b);

        /\* toString() method -> to convert Wrapper object or primitive to String \*/

        Integer I5 = new Integer(10);

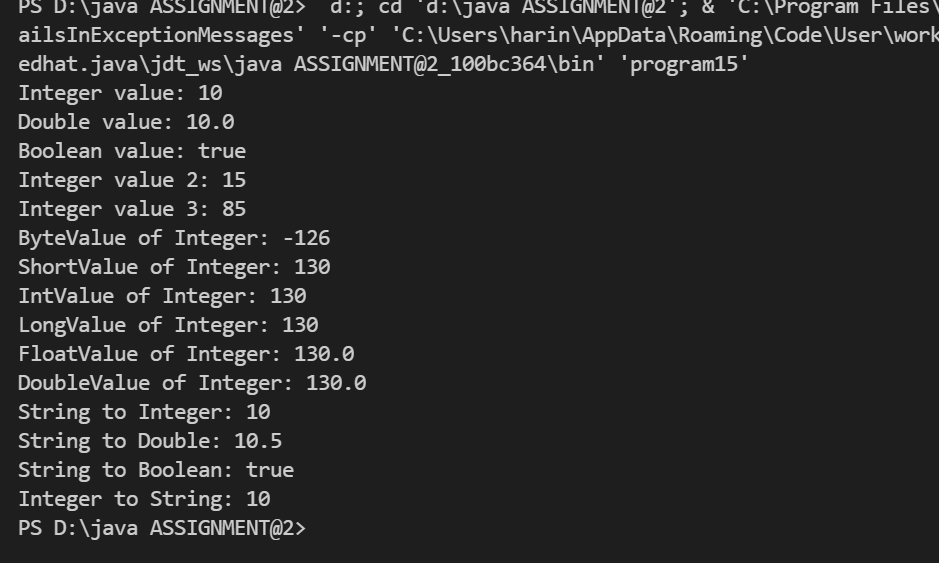
        String s = I5.toString();

        System.out.println("Integer to String: " + s);

    }

}

**OUTPUT**



**Q.16- Write a Java Program to implement inheritance and demonstrate the use of method overriding.**

class Animal {

   public void displayInfo() {

      System.out.println("I am an animal.");

   }

}

class Dog extends Animal {

   @Override

   public void displayInfo() {

      System.out.println("I am a dog.");

   }

}

class program16{

   public static void main(String[] args) {

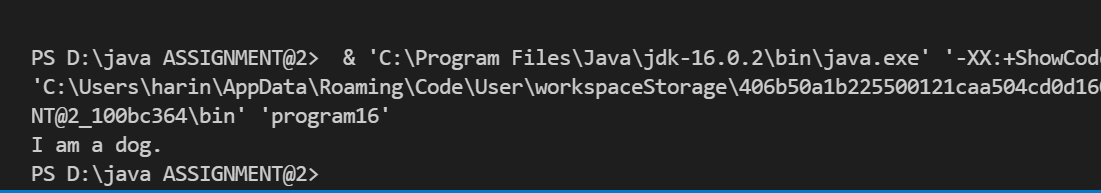
      Dog d1 = new Dog();

      d1.displayInfo();

   }

}

**OUTPUT**



**Q.17- Write a program to demonstrate the use of implementing extending interfaces.**

interface ParentInterface {

    void parentMethod();

}

interface ChildInterface extends ParentInterface {

    void childMethod();

}

class ImplementingClass implements ChildInterface {

    public void childMethod() {

        System.out.println("Child Interface method!!");

    }

    public void parentMethod() {

        System.out.println("Parent Interface mehtod!");

    }

}

class ExtendingAnInterface {

    public static void main(String[] args) {

        ImplementingClass obj = new ImplementingClass();

        obj.childMethod();

        obj.parentMethod();

    }

}

**OUTPUT**

