**NAME-NIKITA VYAS ROLLNO.-233054**

**Concepts of Programming**

**Assignment**

**DAY 1 Assignment: -**

**1: write program to test Hello World.**

**Codes:** -

Package day1;

**public** **class** Hellojava {

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

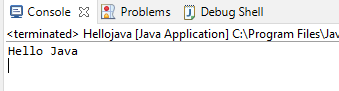
{

System.***out***.println("Hello Java");

}

}

**Output: -**



**2: Write a program to addition of two numbers.**

**Codes:** -

**package** day1;

**import** java.util.Scanner;

**public** **class** add2num {

Scanner sc;

**int** n1,n2,n3;

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

{

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

**int** no1,no2,no3;

System.***out***.println("Enter No1:");

no1=sc.nextInt();

System.***out***.println("Enter No2:");

no2=sc.nextInt();//

no3=no1+no2;

System.***out***.println("Add="+no3);

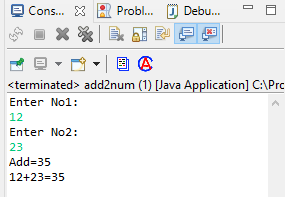
System.***out***.println(no1+"+"+no2+"="+no3);//

sc.close();

}

}

**Output: -**



**3: Write a program to swap two numbers.**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** swap {

Scanner sc;

**int** n1,n2,n3;

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

{

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

**int** n1,n2,temp;

System.***out***.println("Enter Number1:");

n1=sc.nextInt();//

System.***out***.println("Enter Number2:");

n2=sc.nextInt();//

temp=n2;

n1=n2;

n2=temp;

System.***out***.println("After Swampping");

System.***out***.println("Enter Number1 is=:"+n1);

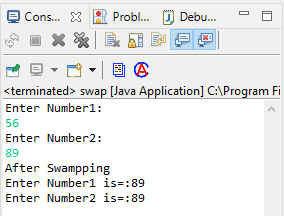
System.***out***.println("Enter Number2 is=:"+n2);

sc.close();

}

}

**Output: -**



**4: Write a program to find factorial of a given number. ex:no5 fact=5\*4\*3\*2\*1=120**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** factorial {

Scanner sc;

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

{

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

**int** n1,i,res=1;

System.***out***.println("Enter Number for factorial:");

n1=sc.nextInt();

**for**( i=1;i<=n1;i++)

{

res=res\*i;

sc.close();

}

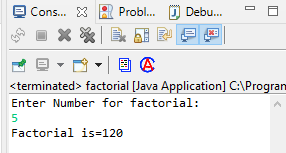
System.***out***.println("Factorial is=" +res);

sc.close();

}

}

**Output: -**



**5: Write a program to find m to the power n. m=3 and n=4 so 3\*3\*3\*3**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** Power {

Scanner sc;

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

{

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

**int** i,m,n,res=1;

System.***out***.println("Enter Number m");

m=sc.nextInt();

System.***out***.println("Enter Number for Power n");

n=sc.nextInt();

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

{

res=res\*m;

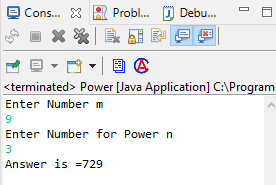
sc.close(); }

System.***out***.println("Answer is ="+res);

}

}

**Output: -**



**6: Check if number is a prime number or not.:**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** prime

{

Scanner sc;

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

{

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

**int** n,i;

**boolean** flag = **false**;

System.***out***.println("Enter Number=");

n=sc.nextInt();

**for** ( i = 2; i <= n / 2; ++i)

{

**if** (n % i == 0)

{

flag = **true**;

**break**;

}

sc.close(); }

**if** (!flag)

System.***out***.println(n + " is a prime number.");

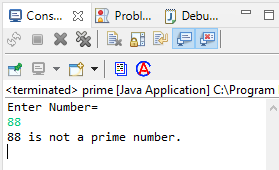
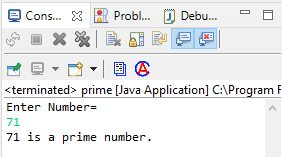
**else**

System.***out***.println(n + " is not a prime number.");

}

}

**Output: -**



**7: Sum of series: 1+2+3+…. +n**

**Codes: -**

**package day1;**

**import** java.util.Scanner;

**public** **class** SumSeries {

Scanner sc;

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

{

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

**int** i,n,k=0;

System.***out***.println("Enter the Number Till sum to Print-");

n=sc.nextInt();

**for**(i=0;i<=n;i++)

{

k=k+i;

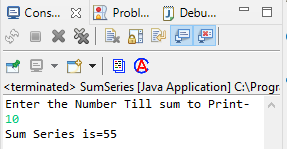
sc.close();}

System.***out***.println("Sum Series is="+k);

}

}

**Output: -**



**8: Check whether the number is palindrome or not?**

**Codes: -**

**package day1;**

**import** java.util.Scanner;

**public** **class** Palindrome {

Scanner sc;

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

{

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

**int** k,c,n,rev=0;

System.***out***.println("Enter Number to check pallindrome ");

n=sc.nextInt();

k=n;

**do**{

c=k%10;

rev=(rev\*10)+c;

k=k/10;

}

**while**(k!=0);

**if**(n==rev)

System.***out***.println("Number is pallindrome ");

**else**

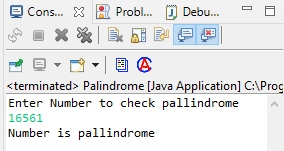
System.***out***.println("Number is not pallindrome ");

sc.close();

}

}

**Output: -**



**9: Write a program to find sum of all even and odd numbers between 1 to n.**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** OddEvenSum

{

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

{

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

**int** i,num,oddSum=0,evenSum=0;

System.***out***.print("Enter the number for num: \n");

num=sc.nextInt();

**for**(i=1; i<=num; i++)

{

**if**(i%2==0)

evenSum=evenSum+i;

**else**

oddSum=oddSum+i;

sc.close();

}

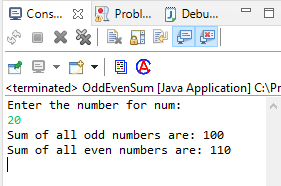
System.***out***.println("Sum of all odd numbers are: "+oddSum);

System.***out***.println("Sum of all even numbers are: "+evenSum);

}

}

**Output:** -



**10: Write a program to enter a number and print its reverse.**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** Reverse {

Scanner sc;

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

{

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

**int** k,c,n,rev=0;

System.***out***.println("Enter Number to Reverse = ");

n=sc.nextInt();

k=n;

**do**

{

c=k%10;

rev=(rev\*10)+c;

k=k/10;

}

**while**(k!=0);

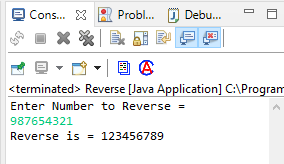
System.***out***.println("Reverse is = "+rev);

sc.close();

}

}

**Output:** -



**11: Write a program to print all Prime numbers between 1 to n.**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** Primetilln

{

Scanner sc;

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

{

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

**int** n1,i,j;

System.***out***.println("Enter Number Till Prime to print ");

n1=sc.nextInt();

System.***out***.println("Primes Number are= ");

**for**(i=2;i<=n1;i++)

{

**int** c=0;

**for**(j=1;j<=i;j++)

{

**if**(i%j==0)

{

c++;

}

}

**if**(c==2)

{

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

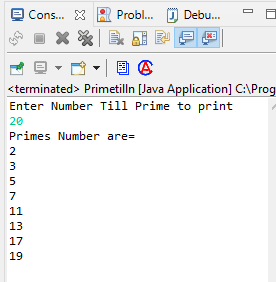
}sc.close();

}

}

}

**Output: -**



**12: Write a program to check entered number is Armstrong number or not.**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** Armsstrong

{

Scanner sc;

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

{

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

**int** k,b,c=0,n;

System.***out***.println("Enter Number to check it is Armsstrong = ");

n=sc.nextInt();

k=n;

**while**(n>0)

{

b=n%10;

c=c+(b\*b\*b);

n=n/10;

}

**if**(k==c)

System.***out***.println("Entered Number is Armsstrong");

**else**

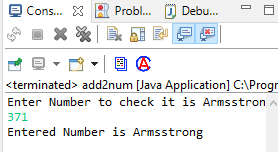
System.***out***.println("Entered Number is not Armsstrong ");

sc.close();

}

}

**Output: -**



**13: Write a program to find greatest of three numbers using nested if-else.**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** GreatestNumber {

Scanner sc;

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

{

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

**int** n1,n2,n3;

System.***out***.println("Enter Number n1 ");

n1=sc.nextInt();

System.***out***.println("Enter Number n2 ");

n2=sc.nextInt();

System.***out***.println("Enter Number n3 ");

n3=sc.nextInt();

**if**(n1 >= n2 && n1 >= n3)

System.***out***.println("Greatest Number is n1="+n1);

**if**(n2 >= n1 && n2 >= n3)

System.***out***.println("Greatest Number is n2 ="+n2);

**if**(n3>=n1 && n3>=n2)

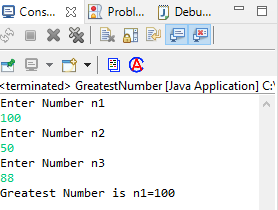
System.***out***.println("Greatest Number is n3= "+n3);

sc.close();

}

}

**Output: -**



**14: Create menu driven program for Pizza Shop.And display total amount,**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** PizzaShope

{

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

// pizza shop

System.***out***.println("1:Large Pizza \n2 :Medium Pizza \n3:Small Pizza \n4: Total \n5:Exit");

**int** ch;

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

**boolean** status = **true**;// assign

**int** qty;

**int** totalAmount = 0;

**while** (status == **true**) {

System.***out***.println("Enter Choice=");

ch = sc.nextInt();// user will enter 1 /2/3/4/5...anything

**switch** (ch) {

**case** 1:

System.***out***.println("Enter Qty for Large Pizza:");

qty = sc.nextInt();

totalAmount += qty \* 450;

**break**;

**case** 2:

System.***out***.println("Enter Qty for Medium :");

qty = sc.nextInt();

totalAmount = totalAmount + (qty \* 250);

**break**;

**case** 3:

System.***out***.println("Enter Qty for Small :");

qty = sc.nextInt();

totalAmount = totalAmount + (qty \* 100);

**break**;

**case** 4:System.***out***.println("Total Amount:"+totalAmount);

**break**;

**case** 5:

status = **false**;

**break**;

**default**:System.***out***.println("INVALID CHOICE") ;

**break**;

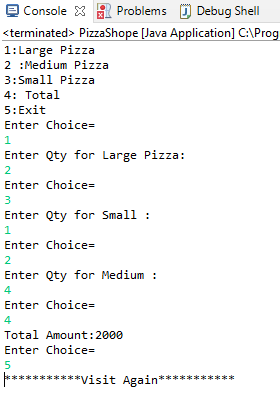
}

}

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*Visit Again\*\*\*\*\*\*\*\*\*\*\*");

sc.close(); }}

**Output: -**



**15: Create Menu driven program for array operations.**

**1: Read Array 2: Print Array 3: Search element in array 4: Reverse Array 5: Even number from array 6: sum of array element**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** Array

{

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

{

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

System.***out***.println("Enter the size of Array : ");

**int** size = sc.nextInt();

**int**[] arr = **new** **int**[size];

System.***out***.println("Enter array elemnts : ");

**for**(**int** i =0;i<size;i++)

{

arr[i] = sc.nextInt();

}

**int** ch;

**do**

{

System.***out***.println("1:Read Array \n2:Print Array \n3:Search element in array \n4:Reverse Array \n5:Even number from array \n6:sum of array element \n7.EXIT");

System.***out***.println("Enter your choice : ");

ch = sc.nextInt();

**switch**(ch)

{

**case** 1:

**for**(**int** i =0;i<size;i++) {

System.***out***.println(arr[i]+" ");}

**case** 2:

**for**(**int** i =0;i<size;i++){

System.***out***.print(arr[i]+" ");

}

**break**;

**case** 3:

System.***out***.println("Search Number in Array");

System.***out***.println("Enter number to search :- ");

**int** num = sc.nextInt();

**boolean** flag = **true**;

**int** i;

**for**( i=0;i<arr.length;i++)

{

**if**(arr[i]==num)

{

flag = **true**;

**break**;

}

**else**

{

flag = **false**;

}

}

**if**(flag==**true**)

{

System.***out***.println("SEARCHED NUBER IS AT "+(i+1)+" position");

}

**else**

{

System.***out***.println("NUMBER NOT FOUND");

}

**break**;

**case** 4:

System.***out***.println("Reverse Array");

**int** temp,start=0,end = size -1;

**while**(start<end)

{

temp = arr[start];

arr[start] = arr[end];

arr[end] = temp;

start++;

end--;

}

System.***out***.println("Reverse Array is : ");

**for**(**int** j=0;j<arr.length;j++) {

System.***out***.println(arr[j]);}

**break**;

**case** 5 :

System.***out***.println("Enter the even numbers of Array : ");

**for**(i=0;i<arr.length;i++) {

**if**(arr[i]%2==0) {

System.***out***.println(arr[i]);

}

}

**break**;

**case** 6 :

System.***out***.println("Sum Of Array Element");

**int** sum = 0;

**for**(i=0;i<arr.length;i++)

{

sum +=arr[i];

}

System.***out***.println(" Sum of arrays =" +sum);

**break**;

**case** 7:

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

**break**;

**default**:

System.***out***.println("Invalid Choice");

**break**;

}

}

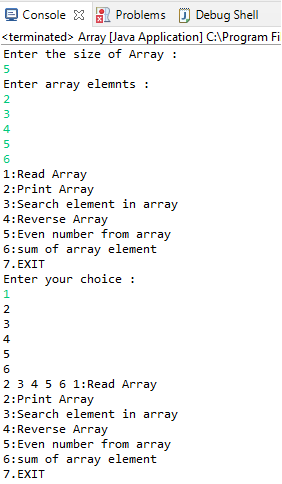
**while**(ch != 7);

sc.close();

}

}

**Output: -**



**16: Read two int array...and store both in third array and display third array**

**arr1: ----->1 2 3**

**arr2:----->5 6 7 8 9**

**result----->1 2 3 5 6 7 8 9**

**Codes: -**

**package** day1;

**import** java.util.Scanner;

**public** **class** AddArray

{

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

{

**int** array1, array2;

**int** arr1[], arr2[], result[];

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

System.***out***.print("Enter the size of the first array : ");

array1 = sc.nextInt();

System.***out***.print("Enter the size of the second array : ");

array2 = sc.nextInt();

arr1 = **new** **int**[array1];

arr2 = **new** **int**[array2];

result = **new** **int**[array1 + array2];

System.***out***.println("Enter elements for the first array :");

**for** (**int** i = 0; i < array1; i++)

{

System.***out***.println("Enter element " + (i + 1) + " : ");

arr1[i] = sc.nextInt();

}

System.***out***.println("Enter elements for the second array :");

**for** (**int** i = 0; i < array2; i++)

{

System.***out***.println("Enter element " + (i + 1) + " : ");

arr2[i] = sc.nextInt();

}

**for** (**int** i = 0; i < array1 + array2; i++)

{

**if** (i < array1)

{

result[i] = arr1[i];

} **else**

{

result[i] = arr2[i - array1];

}

}

System.***out***.println("Final array after merged : ");

**for** (**int** i = 0; i < array1 + array2; i++)

{

System.***out***.print(result[i] + " ");

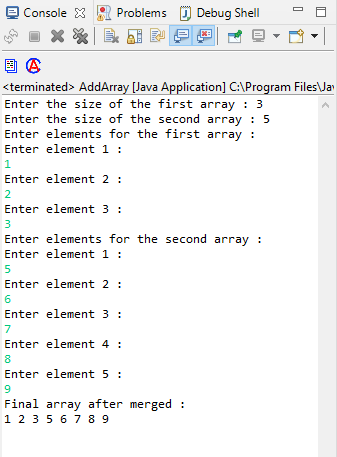
sc.close();

}

}

}

**Output: -**



**Day 2-3 Assignment: -**

**1: Create Date Class with Data Members day, month, year. Write no argument and parameterised constructor .Create two object s and initialize them using no argument and parameterised constructor respectively.**

**1.1: Create an object and initialize it using set Date methods and display it using display Date methods.**

**Codes: -**

**package** day2;

**public** **class** Date {

**private** **int** day;

**private** **int** month;

**private** **int** year;

**public** **void** displayDate()

{

System.***out***.println("Date:"+day+"/"+month+"/"+year);

}

**public** **void** setDate(**int** d, **int** m, **int** y)

{

day = d;

month = m;

year = y;

}

}

**TestDate: -**

**package** day2;

**public** **class** TestDate {

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

Date today = **new** Date();

today.displayDate();

Date tmrw = **new** Date();

tmrw.setDate(12, 03, 2022);

tmrw.displayDate();

Date present = **new** Date();

present.setDate(11, 03, 2022);

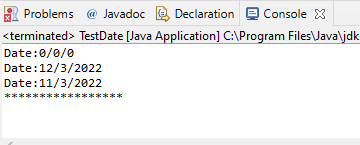
present.displayDate();

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

}

}

**Output: -**



2. **2:Create Employee class with members id(int),name(string),dob(Date).Use above created Date class.**

**Write default and parameterised constructor in Employee Class.Write accept() function to accept information and display() to display emp information.**

class Date {

private int date;

private int mm;

private int yy;

public Date() {

date = 20;

mm = 03;

yy = 2023;

}

public Date(int date, int mm , int yy)

{

this.date = date;

this.mm = mm;

this.yy = yy;

}

@Override

public String toString() {

return "Date [date=" + date + ", mm=" + mm + ", yy=" + yy + "]";

}

public void disp() {

System.out.println("Date = "+date+"mm = "+mm+ "yy = "+yy);

}

}

class Employee {

private int id;

private String name;

Date dob = new Date(25,12,2000);

public Employee() {

this.id = 43;

this.name = "Niki";

}

public Employee(int id, String name) {

this.id = id;

this.name = name;

}

public void display() {

System.out.println("ID = "+this.id+" Name = "+this.name+ " Dob = "+dob);

}

}

public class Day3\_2 {

public static void main(String[] args) {

Employee emp = new Employee();

emp.display();

}

}

**Output:**

**ID = 43 Name = Niki Dob = Date [date=25, mm=12, yy=2000]**

**3. Create a class Person with data members as name, age, city. Write getters and setters for all the data members. Also add the display function. Create Default and Parameterized constructors. Create the object of this class in main method and invoke all the methods in that class.**

**Codes: -**

**package** Day4;

**public** **class** Person {

**private** String name;

**private** **int** age;

**private** String city;

**public** Person() {

name = "Amit";

age=25;

city="Delhi";

}

**public** Person (String name, **int** age, String ct) {

**this**.name = name;

**this**.age = age;

city = ct;

}

**public** **void** display() {

System.***out***.println("General details of a person = ");

System.***out***.println("\nName = "+name+"\nAge = "+age+"\nCity = "+city);

System.***out***.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

**public** **void** setName(String newname) {

**this**.name = newname;

}

**public** **void** setAge(**int** newage) {

**this**.age = newage;

}

**public** **void** setCity(String newcity) {

**this**.city = newcity;

}

**public** String getName() {

**return** **this**.name;

}

**public** **int** getAge() {

**return** **this**.age;

}

**public** String getCity() {

**return** **this**.city;

}

}

**TestPerson: -**

**package** Day4;

**public** **class** TestPerson {

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

Person p1 = **new** Person();

p1.display();

Person p2 = **new** Person("nikita", 30, "jodhpur");

p2.display();

p1.setName("ABC");

p1.setAge(24);

p1.setCity("PUNE");

p1.display();

p2.setName("XYZ");

p2.setAge(26);

p2.setCity("GOA");

p2.display();

String xyz = p1.getName();

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

**int** ds = p1.getAge();

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

String hj = p1.getCity();

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

String lol = p2.getName();

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

**int** sd = p2.getAge();

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

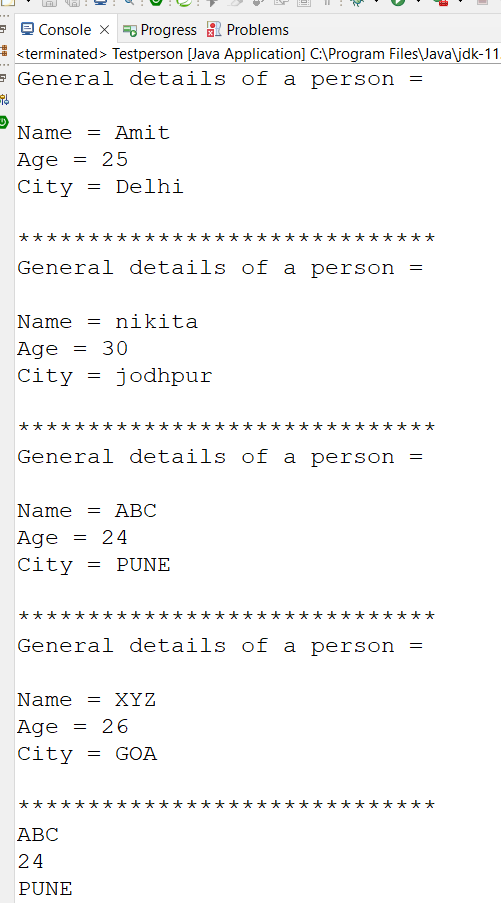
String jh= p2.getCity();

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

}

}

**Output: -**



**4**

**package** Day4;

**public** **class** Book {

**private** **int** id;

**private** String bname,author;

**private** **double** price;

**public** Book() {

bname = "OOP";

id = 707;

author = "ABC";

price = 450;

}

**public** Book(String bname, **int** id, String author, **double** price) {

**this**.bname = bname;

**this**.id = id;

**this**.author = author;

**this**.price = price;

}

**public** **void** display() {

System.***out***.println("BOOK DETAILS :- ");

System.***out***.println("Book Name = "+bname+" \nBook ID = "+id+"\nAuthor Name ="+author+"\nPrice = "+price);

}

**public** **void** setBname(String newname) {

**this**.bname = newname;

}

**public** **void** setId(**int** newid) {

**this**.id = newid;

}

**public** **void** setAuthor(String newnamee) {

**this**.author = newnamee;

}

**public** String getBname() {

**return** **this**.bname;

}

**public** **int** getId() {

**return** **this**.id;

}

**public** String getAuthor() {

**return** **this**.author;

}

}

**TestBook: -**

**package** Day4;

**import** java.util.Scanner;

**public** **class** TestBook

{

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

{

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

System.***out***.println("Enter book name");

String bname = sc.next();

Book b1 = **new** Book();

b1.display();

Book b2 = **new** Book(bname,909,"PQR",1000);

b2.display();

b1.setBname("DBMS");

b1.setId(850);

b1.setAuthor("XYZ");

b1.display();

String xz = b1.getBname();

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

**int** po = b1.getId();

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

String sir = b1.getAuthor();

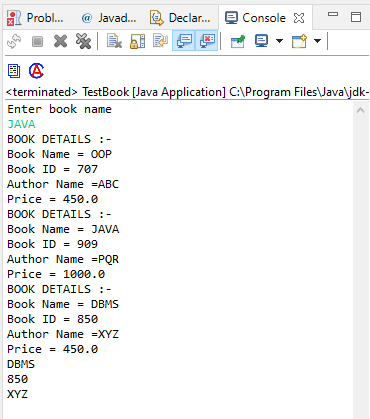
System.***out***.println(sir);

sc.close();

}

}

Output



**5. Create a class Point with data members as x,y. Create Default and Parameterized constructors. Write**

**getters and setters for all the data members. Also add the display function. Create the object of this**

**class in main method and invoke all the methods in that class.**

**Codes: -**

**package** Day4;

**public** **class** Point {

**private** **double** x;

**private** **long** y;

**public** Point() {

}

**public** Point(**double** x, **long** y) {

**this**.x = x;

**this**.y=y;

}

**public** **void** display() {

System.***out***.println("POINT CORDINATES ARE :- ");

System.***out***.println("X-cordinate - "+x+" Y-cordinate - "+y);

}

**public** **void** setX(**double** newx) {

**this**.x = newx;

}

**public** **void** setY(**long** newy) {

**this**.y = newy;

}

**public** **double** getX() {

**return** **this**.x;

}

**public** **long** getY() {

**return** **this**.y;

}

}

**TestPoint: -**

**package** Day4;

**public** **class** TestPoint {

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

Point p2 = **new** Point(10, 5);

p2.display();

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Setter\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

p2.setX(7);

p2.setY(10);

p2.display();

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Getter\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

**double** xc = p2.getX();

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

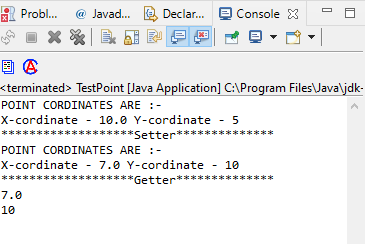
**long** df = p2.getY();

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

}

}

**Output: -**



**6. Create a class ComplexNumber with data members real, imaginary. Create Default and Parameterized constructors. Write getters and setters for all the data members. Also add the display function. Create the object of this class in main method and invoke all the methods in that class.**

**Codes: -**

Public class **package** Day4;

**public** **class** ComplexNumber {

**private** **double** real;

**private** **long** imaginary;

**public** ComplexNumber() {

}

**public** ComplexNumber(**double** real, **long** imaginary) {

**this**.real = real;

**this**.imaginary=imaginary;

}

**public** **void** display() {

System.***out***.println("Complex Numbers are :- ");

System.***out***.println("Real number - "+real+" Imaginary number - "+imaginary);

}

**public** **void** setReal(**double** newx) {

**this**.real = newx;

}

**public** **void** setImaginary(**long** newy) {

**this**.imaginary = newy;

}

**public** **double** getReal() {

**return** **this**.real;

}

**public** **long** getImaginary() {

**return** **this**.imaginary;

}

}

Test ComplexNumber: -

**package** Day4;

**public** **class** TestComplexNumber {

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

ComplexNumber p1 = **new** ComplexNumber();

p1.display();

ComplexNumber p2 = **new** ComplexNumber(10, 5);

p2.display();

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Setter\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

p2.setReal(7);

p2.setImaginary(10);

p2.display();

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Getter\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

**double** xc = p2.getReal();

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

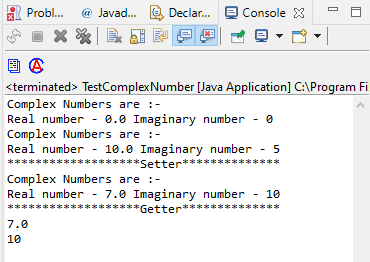
**long** df = p2.getImaginary();

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

}

}

Output: -



**Day 4 Assignment: -**

**1:Write a program to create student class with data members rollno, marks1,mark2,mark3.**

**Accept data (acceptInfo()) and display using display member function.**

**Also display total,percentage and grade.**

**Codes: -**

**package** Day4;

**public** **class** Student {

**private** String name;

**private** **int** rollno, marks1,marks2,marks3;

**private** **double** totalmarks,percentage;

**private** **char** grade;

**public** **void** calculations() {

totalmarks = marks1+marks2+marks3;

percentage = (totalmarks/300)\*100;

}

**public** **void** accept(String nm, **int** rollno, **int** m1,**int** m2,**int** m3) {

}

**public** Student() {

System.***out***.println("Marks obtained out of 100 -: ");

}

**public** **void** calcGrade() {

**if**(percentage>=85) {

grade = 'A';

}

**else** **if**(percentage<85 && percentage >=70) {

grade = 'B';

}

**else** **if**(percentage <70 && percentage >=60) {

grade = 'C';

}

**else** {

grade = 'D';

}

}

**public** Student(String nm, **int** rollno, **int** m1,**int** m2,**int** m3) {

name = nm;

**this**.rollno=rollno;

marks1=m1;

marks2=m2;

marks3=m3;

}

**public** **void** display() {

calculations();

calcGrade();

System.***out***.println("\nMarks obtain in ENG = "+marks1);

System.***out***.println("\nMarks obtain in PHYSICS = "+marks2);

System.***out***.println("\nMarks obtain in MATHS = "+marks3);

System.***out***.println("\nHere is the full details of the Student -: ");

System.***out***.print("\nName = "+name+"\nRoll Number = "+rollno+" \nTotal Marks = "+totalmarks+"\nPercentage = "+percentage+"\nGrade = "+grade );

System.***out***.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

**Test Student: -**

**package** Day4;

**import** java.util.Scanner;

**public** **class** TestStudent

{

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

{

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

System.***out***.println("Enter the Roll Number of ABC :- ");

**int** rollno = sc.nextInt();

System.***out***.println("Enter the Marks of ABC :- ");

Student std1 = **new** Student("ABC",rollno,sc.nextInt(),sc.nextInt(),sc.nextInt());

std1.display();

System.***out***.println("Enter the Roll Number of PQR :- ");

**int** rolln = sc.nextInt();

System.***out***.println("Enter the Marks of PQR :- ");

Student std2 = **new** Student("PQR",rolln,sc.nextInt(),sc.nextInt(),sc.nextInt());

std2.display();

System.***out***.println("Enter the Roll Number XYZ :- ");

**int** roll = sc.nextInt();

System.***out***.println("Enter the Marks of XYZ :- ");

Student std3 = **new** Student("XYZ",roll,sc.nextInt(),sc.nextInt(),sc.nextInt());

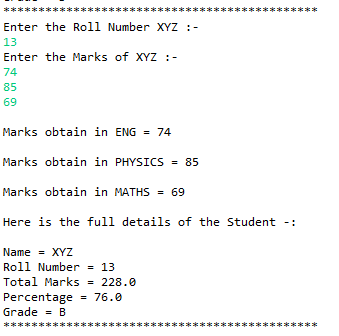
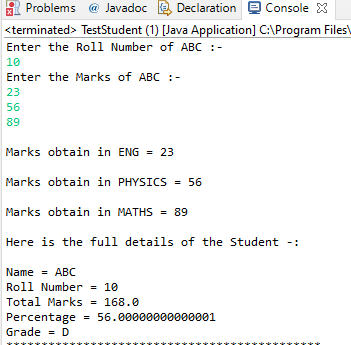
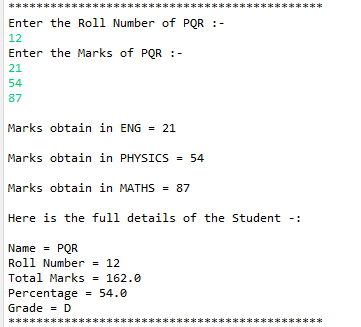
std3.display();

sc.close();

}

}

**Output: -**



**4. Create a class Point with data members as x,y. Create Default and Parameterized constructors. Write**

**getters and setters for all the data members. Also add the display function. Create the object of this**

**class in main method and invoke all the methods in that class.**

**Codes: -**

**package** Day4;

**public** **class** Point {

**private** **double** x;

**private** **long** y;

**public** Point() {

}

**public** Point(**double** x, **long** y) {

**this**.x = x;

**this**.y=y;

}

**public** **void** display() {

System.***out***.println("POINT CORDINATES ARE :- ");

System.***out***.println("X-cordinate - "+x+" Y-cordinate - "+y);

}

**public** **void** setX(**double** newx) {

**this**.x = newx;

}

**public** **void** setY(**long** newy) {

**this**.y = newy;

}

**public** **double** getX() {

**return** **this**.x;

}

**public** **long** getY() {

**return** **this**.y;

}

}

**TestPoint: -**

**package** Day4;

**public** **class** TestPoint {

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

Point p2 = **new** Point(10, 5);

p2.display();

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Setter\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

p2.setX(7);

p2.setY(10);

p2.display();

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Getter\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

**double** xc = p2.getX();

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

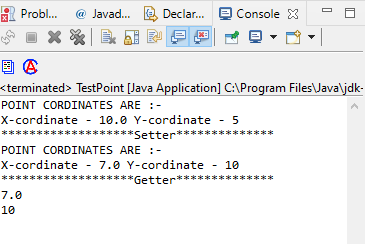
**long** df = p2.getY();

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

}

}

**Output: -**



**5. Create a class ComplexNumber with data members real, imaginary. Create Default and Parameterized constructors. Write getters and setters for all the data members. Also add the display function. Create the object of this class in main method and invoke all the methods in that class.**

**Codes: -**

Public class **package** Day4;

**public** **class** ComplexNumber {

**private** **double** real;

**private** **long** imaginary;

**public** ComplexNumber() {

}

**public** ComplexNumber(**double** real, **long** imaginary) {

**this**.real = real;

**this**.imaginary=imaginary;

}

**public** **void** display() {

System.***out***.println("Complex Numbers are :- ");

System.***out***.println("Real number - "+real+" Imaginary number - "+imaginary);

}

**public** **void** setReal(**double** newx) {

**this**.real = newx;

}

**public** **void** setImaginary(**long** newy) {

**this**.imaginary = newy;

}

**public** **double** getReal() {

**return** **this**.real;

}

**public** **long** getImaginary() {

**return** **this**.imaginary;

}

}

Test ComplexNumber: -

**package** Day4;

**public** **class** TestComplexNumber {

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

ComplexNumber p1 = **new** ComplexNumber();

p1.display();

ComplexNumber p2 = **new** ComplexNumber(10, 5);

p2.display();

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Setter\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

p2.setReal(7);

p2.setImaginary(10);

p2.display();

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Getter\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

**double** xc = p2.getReal();

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

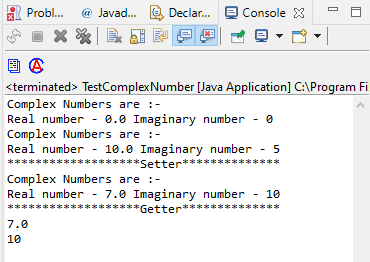
**long** df = p2.getImaginary();

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

}

}

Output: -



**Ques 2-5 same as previous.**

**7:Create Student class with rollno,name address.**

**Write business logic for auto incrment of rollno(don't accept roll no from user)**

**Write parameterised constr for accepting name and address only**

**Write getter and setter and display function**

**7.1 Test Student class by creating 5 diff object.and display aal details(chk rollno created automatically)**

**7.2 Create an array of 5 students and show only names**

**7.3 modify 1.2 for showing students details who lives in Pune**

**Codes: -**

**package** Day6;

**public** **class** Student {

**private** String name, address;

**private** **int** rollno;

**private** **int** count;

**private** **static** **int** *counter*;

**private** **static** **int** *rollnoCounter*;

**static** {

System.***out***.println("----static-----");

*rollnoCounter* = 2200100;

*counter* =100;

}

**public** Student() {

**this**.name = name;

**this**.address = address;

}

**public** Student(String name, String address) {

**this**.name = name;

**this**.address = address;

*rollnoCounter* = *rollnoCounter*+1;

rollno = *rollnoCounter*

*counter* = *counter* +1;

count = *counter*;

}

**public** **void** display() {

System.***out***.println("Student Details : name - "+name+"; address - "+address+"; rollno - "+rollno+ "; "+count);

}

**public** **void** setName(String newname) {

**this**.name = newname;

}

**public** **void** setAddress(String newadd) {

**this**.name = newadd;

}

**public** String getName(){

**return** **this**.name;

}

**public** String getAddress() {

**return** **this**.address;

}

}

**TestStudent: -**

**package** Day6;

**import** java.util.Scanner;

**public** **class** TestStudent {

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

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

Student std1 = **new** Student("jin","pune");

std1.display();

Student std2 = **new** Student("ram","himchl");

std2.display();

Student std3 = **new** Student("shyam","jaipur");

std3.display();

Student std4 = **new** Student("hisname","hisplace");

std4.display();

Student std5 = **new** Student("hername","pune");

std5.display();

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

Student[] arr = **new** Student[5];

**for**(**int** i =0; i<arr.length;i++) {

System.***out***.println("Enter the student name and address ");

arr[i]= **new** Student(sc.next(),sc.next());

//arr[i].display();

}

**for**(**int** i =0;i<5;i++) {

arr[i].display();

}

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

System.***out***.println("SHOW name of all students : ");

**for** (**int** j =0;j<5;j++) {

System.***out***.println(arr[j].getName());

}

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

System.***out***.println("Students living in pune only : ");

**for**( **int** k =0;k<5;k++) {

**if**(arr[k].getAddress().equals("pune"))

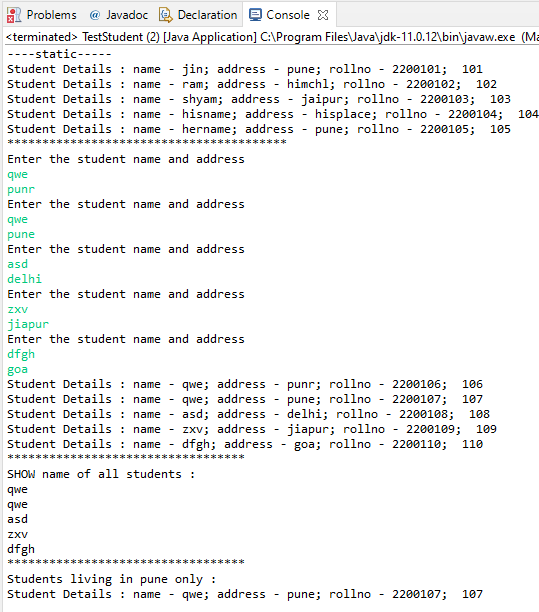
arr[k].display();

}

}

}

**Output: -**



**8:Create diff package and add class inside that.Try to access one package class in another package....(check default access specifier)**

**Codes: -**

**package** Day6;

**import** pkgDay4.Date;

**public** **class** TestDate {

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

Date day1 = **new** Date();

day1.display();

Date day2 = **new** Date(17,8,1995);

day2.display();

day1.setDay(31);

day1.setMonth(11);

day1.setYear(1984);

day1.display();

day2.setDay(13);

day2.setMonth(10);

day2.setYear(1934);

day2.display();

**int** lol = day1.getDay();

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

**int** pop = day1.getMonth();

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

**int** ioi = day1.getYear();

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

**int** hh = day2.getDay();

System.***out***.print(hh);

**int** tt = day2.getMonth();

System.***out***.print(tt);

**int** yy = day2.getYear();

System.***out***.print(yy);

}

}

**TestDate:-**

**package** day6;

**import** pkg.Day4.Book;

**public** **class** Date {

**int** day;

**private** **int** month,year;

**public** Date() {

}

**public** Date(**int** day, **int** month, **int** year) {

**this**.day = day;

**this**.month = month;

**this**.year=year;

}

**public** **void** display() {

System.***out***.println("Date :: "+day+"/"+month+"/"+year);

}

**public** **void** setDay(**int** newday) {

**this**.day=newday;

}

**public** **void** setMonth(**int** newmonth) {

**this**.month = newmonth;

}

**public** **void** setYear(**int** newyear) {

**this**.year = newyear;

}

**public** **int** getDay() {

**return** **this**.day;

}

**public** **int** getMonth() {

**return** **this**.month;

}

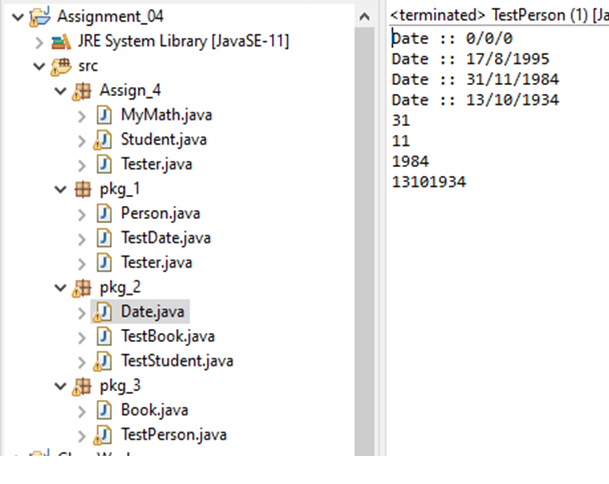
**public** **int** getYear() {

**return** **this**.year;

}

}

**Output: -**



**\* Task 1**

**Create a class Point2D , under package "com.cdac.geometry" for representing a point in x-y co-ordinate system.**

**1.1 Create a parameterized constructor to accept x & y co-ords.**

**1.2 Add public String show()) --to return point's x & y co-ords**

**1.3 Add isEqual method to Point2D class : boolean returning method : must return true if both points are having same x,y co-ords or false otherwise.**

**1.4 Add a method , calculateDistance , to calc distance between 2 points**

**Hint : use distance formula.**

**1.5 Create a driver class(for UI) , in the "tester" package "TestPoints" , with main(..)**

**Ask user , how many points to plot? :**

**Create suitable array.**

**1.6 Accept x,y co-ordinates for all the points n store it suitably.**

**1.7 Display x,y co-ordinates of all the points plotted so far ,using for-each loop.**

**1.8 Accept 2 indices from user .**

**Find out if the points at these indices are same or different (Hint : isEqual)**

**Print the message accordingly.**

**If points are not same , display distance between these 2 points.**

**\*/**

package com.cdac.geometry;

public class Point2D {

private int x;

private int y;

public Point2D() {

super();

this.x = 1;

this.y = 1;

}

public Point2D(int x, int y) {

super();

this.x = x;

this.y = y;

}

public String toString() {

return "x= " + this.x + "y= " + this.y;

}

public boolean CoEqual(Point2D p) {

if (this.x > 0 && this.y > 0 && p.x > 0 && p.y > 0)

return true;

if (this.x < 0 && this.y < 0 && p.x < 0 && p.y < 0)

return true;

if (this.x > 0 && this.y < 0 && p.x > 0 && p.y < 0)

return true;

else if (this.x < 0 && this.y > 0 && p.x < 0 && p.y > 0)

return true;

else

return false;

}

//Add a method , calculateDistance , to calc distance between 2 points

//Hint : use distance formula.

public double calDistance(Point2D p) {

double dis = Math.sqrt(((this.x - p.x) \* (this.x - p.x)) + ((this.y - p.y) \* (this.y - p.y)));

return dis;

}

public void accept(int x, int y) {

this.x = x;

this.y = y;

}

public void display() {

System.out.println("(x,y) = (" + x + "," + y + ")");

}

public void Equal(Point2D p) {

boolean b = this.CoEqual(p);

if (b == true) {

if (this.x == p.x && this.y == p.y)

System.out.println("both x and y are equal ");

else

System.out.println(" x and y are not equal but in same quadrant");

}

else

System.out.println(" x and y are not equal");

}

}

=============================================================

package com.tester;

import java.util.Scanner;

import com.cdac.geometry.Point2D;

public class TestPoint {

public static void main(String[] args) {

System.out.println("how many point create..");

Scanner sc = new Scanner(System.in);

int size = sc.nextInt();

Point2D[] p = new Point2D[size];

// p[0]=new Point2D(1,2);

// p[1]=new Point2D(-1,2);

// p[2]=new Point2D(1,-2);

// p[3]=new Point2D(-1,-2);

// p[4]=new Point2D(8,2);

//

System.out.println("enter points(x,y) " + size);

for (int i=0;i<p.length;i++) {

p[i]=new Point2D();

System.out.print("Enter x and y for point "+(i+1)+": ");

p[i].accept(sc.nextInt(), sc.nextInt());

//System.out.println();

p[i].display();

}

System.out.println("Compare two points of Point2D");

//for(int i=0;i<p.length;i++) {

int n1 = sc.nextInt();

int n2 = sc.nextInt();

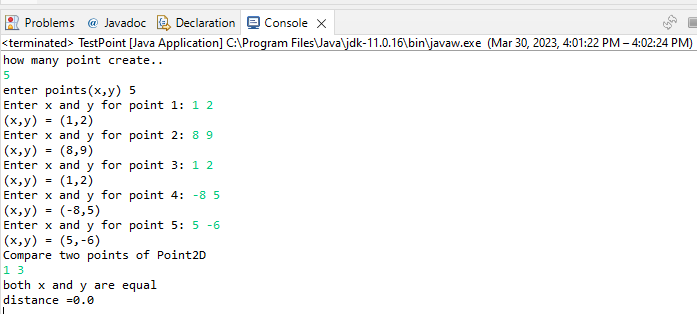
p[n1-1].Equal(p[n2-1]);

double res=p[n1-1].calDistance(p[n2-1]);

System.out.println("distance ="+res);

}

}



**2.1 Can you arrange Fruit,Apple,Orange,Mango in inheritance hierarchy ?**

**Use tight encapsulation.**

**2.2 Properties (instance variables) : color : String , weight : double , name:String, fresh : boolean**

**2.3 Add suitable constructor.**

**2.4 Override toString correctly to return state of all fruits (return only : name ,color , weight )**

**2.5 Add a taste() method to return String form of the taste of the Fruit**

**eg : public String taste()**

**For Fruit : Can you identify taste of any general fruit ?**

**So return : "no specific taste"**

**Apple : should return "sweet n sour"**

**Mango : should return "sweet"**

**Orange : should return "sour"**

**2.6 Add specific functionality , in the sub classes**

**In Mango : public void pulp() {Display name n color of the fruit + a mesg creating pulp!}**

**In Orange : public void juice() {Display name n weight of the fruit + a mesg extracting juice!}**

**In Apple : public void jam() {Display name of the fruit + a mesg making jam!}**

**2.7 Add all of above classes under the package "com.app.fruits"**

**2.8 Create java application FruitBasket , with main method , as a tester**

**2.9 Prompt user for the basket size n create suitable data structure**

**2.10 Supply options**

**1. Add Mango**

**2. Add Orange**

**3. Add Apple**

**NOTE : You will be adding a fresh fruit in the basket , in all of above options.**

**4. Display names of all fruits in the basket.**

**5. Display name,color,weight , taste of all fresh fruits , in the basket.**

**6. Mark a fruit in a basket , as stale(=not fresh)**

**i/p : index**

**o/p : error message (in case of invalid index) or mark it stale**

**7. Mark all sour fruits stale**

**Hint : Use equals() method of the String class.**

**8. Invoke fruit specific functionality (pulp / juice / jam)**

**i/p : index**

**Invoke correct functionality (pulp / juice / jam)**

**10. Exit**

=================================================

package Fruit.shope;

public class Fruit {

//2.2 Properties (instance variables) : color : String , weight : double , name:String, fresh : boolean

private String name;

private String color;

private double weight;

private boolean fresh;

public Fruit(String name, String color, double weight, boolean fresh) {

super();

this.name = name;

this.color = color;

this.weight = weight;

this.fresh = fresh;

}

@Override

public String toString() {

return "Fruit [name=" + name + ", color=" + color + ", weight=" + weight + "]";

}

//2.5 Add a taste() method to return String form of the taste of the Fruit

//eg : public String taste()

//

//For Fruit : Can you identify taste of any general fruit ?

//So return : "no specific taste"

//

//Apple : should return "sweet n sour"

//Mango : should return "sweet"

//Orange : should return "sour"

//

public String taste() {

return "no specific taste ";

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

public double getWeight() {

return weight;

}

public void setWeight(double weight) {

this.weight = weight;

}

public boolean isFresh() {

return fresh;

}

public void setFresh(boolean fresh) {

this.fresh = fresh;

}

========================================================================

package Fruit.shope;

public class Apple extends Fruit {

public Apple(String name, String color, double weight, boolean fresh) {

super(name, color, weight, fresh);

// TODO Auto-generated constructor stub

}

@Override

public String taste() {

return "sweet n sour";

}

public void jam()

{

System.out.println(super.getName()+" making jam !!");

}

}

package Fruit.shope;

public class Mango extends Fruit {

public Mango(String name, String color, double weight, boolean fresh) {

super(name, color, weight, fresh);

// TODO Auto-generated constructor stub

}

@Override

public String taste() {

return "sweet";

}

public void pulp()

{

System.out.println(super.getName()+"creating pulp!");

}

}

package Fruit.shope;

public class Orange extends Fruit{

public Orange(String name, String color, double weight, boolean fresh) {

super(name, color, weight, fresh);

// TODO Auto-generated constructor stub

}

@Override

public String taste() {

return "sour";

}

public void juice()

{

System.out.println(super.getName()+" extracting juice!");

}

package Fruite.tester;

import java.util.Scanner;

import Fruit.shope.Fruit;

import Fruit.shope.\*;

public class FruiteBasket {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int size;

System.out.println("Entr a size of Fruite Basket :");

size = sc.nextInt();

Fruit basket[] = new Fruit[size];

System.out.println("Enter a choice :");

System.out.println("1. Add Mango");

System.out.println("2. Add Orange");

System.out.println("3. Add Apple");

System.out.println("4. Display names of all fruits in the basket.");

System.out.println("5. Display name,color,weight , taste of all fresh fruits , in the basket.");

System.out.println("6. Mark a fruit in a basket , as stale(=not fresh)");

System.out.println("7. Mark all sour fruits stale ");

System.out.println("8. Invoke fruit specific functionality (pulp / juice / jam)");

System.out.println("9. Exit");

System.out.println("");

System.out.println("");

int choice;

int i = 0;

do {

System.out.println("Enter a choice :");

choice = sc.nextInt();

switch (choice) {

case 1:// 1. Add Mango //name, color, weight, fresh)

if (i < basket.length) {

System.out.println("enter a mango(name,color,weight,fresh):");

basket[i] = new Mango(sc.next(), sc.next(), sc.nextDouble(), sc.nextBoolean());

i++;

System.out.println("mango in a basket");

} else

System.out.println("your basket full ");

break;

case 2:// 2. Add Orange

if (i < basket.length) {

System.out.println("enter a Orange(name,color,weight,fresh):");

basket[i] = new Orange(sc.next(), sc.next(), sc.nextDouble(), sc.nextBoolean());

i++;

System.out.println("orange in a basket");

} else

System.out.println("your basket full ");

break;

case 3:// 3. Add Apple

if (i < basket.length) {

System.out.println("enter a Apple(name,color,weight,fresh):");

basket[i] = new Apple(sc.next(), sc.next(), sc.nextDouble(), sc.nextBoolean());

i++;

System.out.println("apple in a basket");

} else

System.out.println("your basket full ");

break;

case 4:// 4. Display names of all fruits in the basket.

for (Fruit F : basket)

if (F != null)

System.out.println(F.getName());

break;

case 5:// 5. Display name,color,weight , taste of all fresh fruits , in the basket."

for (Fruit F : basket) {

if (F != null) {

System.out.println(F);

System.out.println(F.taste());

}

}

break;

case 6:// 6. Mark a fruit in a basket , as stale(=not fresh) System.out.println("enter a index of basket (start with 0):");

int index = sc.nextInt();

for (int j = 0; j < i; j++) {

if (index == j) {

if (basket[j].isFresh() == false)

System.out.println(basket[j].getName() + " is not fresh ");

}

}

break;

case 7:// 7. Mark all sour fruits stale

for(int j=0;j<i;j++)

if(basket[j] instanceof Orange

basket[j].setFresh(false);

break;

case 8:// 8. Invoke fruit specific functionality (pulp / juice / jam)

System.out.println("enter a index of basket: ");

int n = sc.nextInt();

if (n < i) {

for (int j = 0; j < basket.length; j++) {

if (j == n) {

if(basket[j] instanceof Apple)

((Apple)basket[j]).jam();

if(basket[j] instanceof Orange)

((Orange)basket[j]).juice();

if(basket[j] instanceof Mango)

((Mango)basket[j]).pulp();

break;

}

}

}

else

System.out.println("wrong input:");

break;

case 9:// exit

System.out.println("visit again !!!");

break;

default:

System.out.println("invalid choice ");

break;

}

} while (choice != 9);

}

}

output :

Entr a size of Fruite Basket :

5

Enter a choice :

1. Add Mango

2. Add Orange

3. Add Apple

4. Display names of all fruits in the basket.

5. Display name,color,weight , taste of all fresh fruits , in the basket.

6. Mark a fruit in a basket , as stale(=not fresh)

7. Mark all sour fruits stale

8. Invoke fruit specific functionality (pulp / juice / jam)

9. Exit

Enter a choice :

1

enter a mango(name,color,weight,fresh):

dasheri yellow 0.500 true

mango in a basket

Enter a choice :

2

enter a Orange(name,color,weight,fresh):

Nagpuriya lightOrange 0.150 true

orange in a basket

Enter a choice :

3

enter a Apple(name,color,weight,fresh):

Kashmiri DarkRed 2.500 true

apple in a basket

Enter a choice :

2

enter a Orange(name,color,weight,fresh):

Orange DarkOrange 10.000 true

orange in a basket

Enter a choice :

4

dasheri

Nagpuriya

Kashmiri

Orange

Enter a choice :

5

Fruit [name=dasheri, color=yellow, weight=0.5]

sweet

Fruit [name=Nagpuriya, color=lightOrange, weight=0.15]

sour

Fruit [name=Kashmiri, color=DarkRed, weight=2.5]

sweet n sour

Fruit [name=Orange, color=DarkOrange, weight=10.0]

sour

Enter a choice :

6

enter a index of basket (start with 0):

2

Enter a choice :

7

Enter a choice :

8

enter a index of basket:

3

Orange extracting juice!

Enter a choice :

9

**3. Solve this.**

**Fresh business scenario to apply inheritance , polymorphism n abstraction to emp based organization scenario.**

**Create Emp based organization structure --- Emp , Mgr , Worker**

**All of above classes must be in --com.app.org**

**3.1 Emp state--- id(int), firstName, lastName , deptId , basic(double)**

**emp id MUST be automatically generated.**

**Behaviour ---1.  get emp details -- using  toString.**

**2. compute net salary**

**3.2 Mgr state  ---id,name,basic,deptId , perfBonus**

**Behaviour ----1. get mgr details :  using toString.**

**2. compute net salary (formula: basic+perfBonus) -- override computeNetSalary**

**3. get performance bonus. --add a new method to return bonus.(getter)**

**3.3 Worker state  --id,name,basic,deptId,hoursWorked,hourlyRate**

**Behaviour---**

**1. get worker details -- :  override toString.**

**2.  compute net salary (formula:  = basic+(hoursWorked\*hourlyRate) --override computeNetSalary**

**3. get hrlyRate of the worker  -- add a new method to return hourly rate of a worker.(getter)**

**Organize classes in inheritance  hierarchy.**

**NOTE : toString method SHOULD NOT include the net salary of the employee**

package com.app.org;

public class Emp {

private int id;

private String firstname;

private String lastname;

private int dept\_id;

private static int count;

protected double basic;

public Emp(String firstname, String lastname, int dept\_id, double basic) {

super();

this.id = ++count;

this.firstname = firstname;

this.lastname = lastname;

this.dept\_id = dept\_id;

this.basic = basic;

}

@Override

public String toString() {

return "Emp [id=" + id + ", firstname=" + firstname + ", lastname=" + lastname + ", dept\_id=" + dept\_id

+ ", Salary =" + basic + "]";

}

public double netSalary() {

System.out.println("Employee Salary =" + basic);

return 0;

}

public int getId() {

return id;

}

public static int getCount() {

return count;

}

}

package com.app.org;

public class Manager extends Emp {

private int pfbonus;

public Manager(String firstname, String lastname, int dept\_id, double basic, int pfbonus) {

super(firstname, lastname, dept\_id, basic);

this.pfbonus = pfbonus;

}

public int getPfbonus() {

return pfbonus;

}

public void setPfbonus(int pfbonus) {

this.pfbonus = pfbonus;

}

@Override

public String toString()

{

return "Manager [" + super.toString() + ", pfbonus=" + pfbonus + "]";

}

public double netSalary() {

double netSal=(super.basic+this.pfbonus);

return netSal;

}

public void comSal(double inc\_amt) {

double comSal= this.netSalary()+inc\_amt;

System.out.println("updated salary of Manager "+this.getId()+" = "+ comSal);

}

}

package com.app.org;

public class Worker extends Emp {

private int h\_worked;

private double h\_rate;

public Worker(String firstname, String lastname, int dept\_id, double basic, int h\_worked, double h\_rate) {

super(firstname, lastname, dept\_id, basic);

this.h\_worked = h\_worked;

this.h\_rate = h\_rate;

}

@Override

public String toString() {

return "Worker [" + super.toString() + ", h\_worked=" + h\_worked + ", h\_rate=" + h\_rate + ", basic="

+ basic + "]";

}

public double netSalary() {

double netSal=(super.basic+ (this.getH\_rate() \* this.getH\_worked()));

return netSal;

}

public int getH\_worked() {

return h\_worked;

}

public void setH\_worked(int h\_worked)

{

this.h\_worked = h\_worked;

}

public double getH\_rate()

{

return h\_rate;

}

public void setH\_rate(double h\_rate)

{

this.h\_rate = h\_rate;

}

public void comSal(double inc\_amt) {

double comSal= this.netSalary()+inc\_amt;

System.out.println("updated salary of worker "+this.getId()+" = "+comSal);

}

}

=

package com.tester;

import java.util.\*;

import com.app.org.\*;

//import static com.app.org.Emp.\*;

public class TestOrg {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of employees in the organization ");

Emp e[] = new Emp[sc.nextInt()];

int i = 0;

int choice;

System.out.println("1.add Manager\n2.add Worker\n3.display Employee details \n4.Update salary");

do {

System.out.println("Enter the choice");

choice = sc.nextInt();

switch (choice) {

case 1:

if (Emp.getCount() < e.length) {

System.out.println("Enter firstname, lastname, dept\_id, basic, pf\_bonus");

e[i] = new Manager(sc.next(), sc.next(), sc.nextInt(), sc.nextDouble(), sc.nextInt());

System.out.println("New Manager Emp created " + e[i].getId());

i++;

} else

System.out.println("No vacancy for Manager");

break;

case 2:

if (Emp.getCount() < e.length) {

System.out.println("Enter firstname, lastname, dept\_id, basic, hourly worked, hourly rate");

e[i] = new Worker(sc.next(), sc.next(), sc.nextInt(), sc.nextDouble(), sc.nextInt(),

sc.nextDouble());

System.out.println("New Worker Emp crated " + e[i].getId());

i++;

} else

System.out.println("No vacancy for Worker");

break;

case 3:

for (Emp x : e) {

if (x != null) {

System.out.println(x);

x.netSalary();

}

}

break;

case 4:

System.out.println("Enter ID");

int nid = sc.nextInt();

for (int j = 0; j < Emp.getCount(); j++) {

if (e[j].getId() == nid) {

if (e[j] instanceof Manager) {

System.out.println("Enter amount to be incremented :");

((Manager) e[j]).comSal(sc.nextDouble());

}

else if (e[j] instanceof Worker) {

System.out.println("Enter amount to be incremented :");

((Worker) e[j]).comSal(sc.nextDouble());

}

}

}

break;

case 5:

System.out.println(" Enter choice from 1 to 4 ");

break;

}

} while (choice < 5);

System.out.println("Thank you for using my organization \n Please visit again");

}

}

DAY 8

**1. Create abstract class Shape --state : x,y**

**Abstract Method --public double area();**

**public String toString() : to ret x & y**

**Why will area() be abstract in Shape class ?????????**

**2. Circle -- x,y,radius**

**Concrete overriding Method --public double area() : ret area of circle**

**public String toString() : ret x, y & radius**

**3. Rectangle -- x,y,w,h**

**Concrete overriding Method --public double area() : ret area of rectangle**

**public String toString() : ret x, y , width & height**

**4. Square-- x,y,side**

**Concrete overriding Method --public double area() : ret area of square**

**public String toString() : ret x, y , side**

**5. Create a ShapeFactory class**

**Add a static method(generateShape) to return randomly generated shape.**

**Hint : random no generator**

**6. Create a Tester . Invoke ShapeFactory's generateShape() method , in a for-loop (5 times)**

**to display details & area of each shape**

package com.abstractClass;

public abstract class Shape {//abstract class

private int x,y;

public Shape(int x, int y) {

super();

this.x = x;

this.y = y;

}

public int getX() {

return x;

}

public void setX(int x) {

this.x = x;

}

public int getY() {

return y;

}

public void setY(int y) {

this.y = y;

}

@Override

public String toString() {

return "Shape [x=" + x + ", y=" + y + "]";

}

//abstract method

public abstract double area();

}

package com.abstractClass;

public class Circle extends Shape{

private double radius;

public Circle(int x, int y,double radius) {

super(x, y);

this.radius=radius;

}

public double getRadius() {

return radius;

}

public void setRadius(double radius) {

this.radius = radius;

}

@Override

public String toString() {

return super.toString()+" Circle+ [radius=" + radius + "]";

}

@Override

public double area() {

double area=3.14\*Math.pow(radius, 2);

return area;

}

}

package com.abstractClass;

public class Rectangle extends Shape{

private double width,height;

public Rectangle(int x, int y,double width,double height) {

super(x, y);

this.width=width;

this.height=height;

}

public double getWidth() {

return width;

}

public void setWidth(double width) {

this.width = width;

}

public double getHeight() {

return height;

}

public void setHeight(double height) {

this.height = height;

}

@Override

public String toString() {

return super.toString() +"Rectangle [width=" + width + ", height=" + height + "]";

}

@Override

public double area() {

double area=(this.height\*this.width);

return area;

}

}

package com.abstractClass;

public class Sqaure extends Shape{

private double side;

public Sqaure(int x, int y,double side ) {

super(x, y);

this.side=side;

}

public double getSide() {

return side;

}

public void setSide(double side) {

this.side = side;

}

@Override

public String toString() {

return super.toString()+"Sqaure [side=" + side + "]";

}

@Override

public double area() {

double area=this.side\*this.side;

return area;

}

}

package com.TestAbstractClass;

import java.util.Scanner;

import com.abstractClass.\*; //all class import

public class ShapeFactory {

public static Shape generateShape() {

Scanner sc=new Scanner(System.in);

double n=Math.random();

//System.out.println(n);

Shape s = null;

if(n>0 && n<0.33) {

System.out.println("enter for Circle : x ,y ,radius");

System.out.println("Enter x y and rad");

s = new Circle(sc.nextInt(),sc.nextInt(),sc.nextDouble());

return s;

}

else if(n<0.67 && n>0.33) {

System.out.println("enter for : x ,y ,radius");

s = new Rectangle(sc.nextInt(),sc.nextInt(),sc.nextDouble(),sc.nextDouble());

return s;

}

else if(n<1.0 && n>0.67) {

s = new Sqaure(sc.nextInt(),sc.nextInt(),sc.nextDouble());

return s;

}

return s;

}

public static void main(String[] args) {

Scanner sc1=new Scanner(System.in);

for (int i =0;i<5;i++) {

Shape res = generateShape();

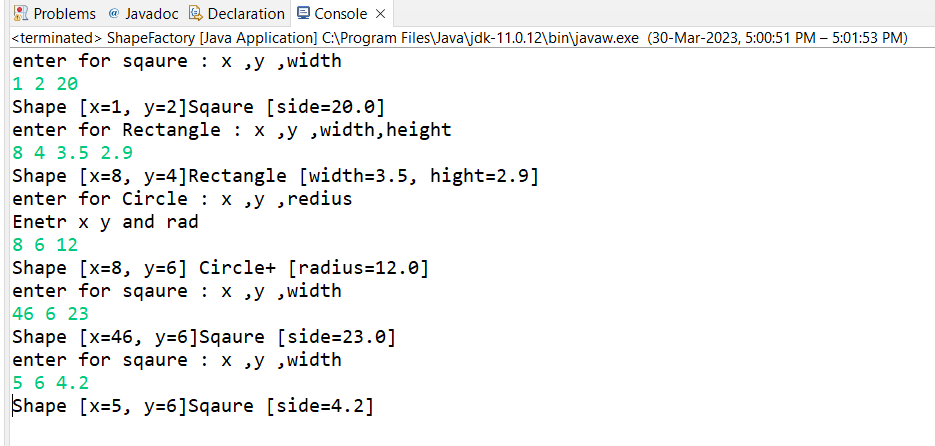
if(res !=null)

System.out.println(res);

}

}

}



**7Objective**

**:Define an interface and implement it in any class wherever it is required.**

**Pre-condition**

**: Employee, Date and Shape class should be created.**

**Problem Statement**

**7.1:Define an interface Printable with a method print(). Implement this**

**interface in Employee, Shape and Date class.**

package com.Interface;

public interface Iprintable {

//all data members are by default public final and static

//int abc=100;

//methods are public abstract

default void print() //no need to write public abstract void print();

{System.out.println("Inside Interface class");}

}

package com.Interface;

public class Date implements Iprintable

{

public void print() {

System.out.println("Print method in Date Class");

}

}

package com.Interface;

public class Emp implements Iprintable

{

@Override

public void print() {

System.out.println("Print method in Manager Class");

}

}

public class Shape implements Iprintable

{

public void print() {

System.out.println("Inside Shape class");

}

}

package com.Interface;

public class TestInterference {

public static void display(Iprintable p)

{

p.print();

}

public static void main(String[] args)

{

Date d = new Date();

Emp e =new Emp();

Shape s = new Shape();

System.out.println("Print function by object creation");

d.print();

e.print();

s.print();

System.out.println("================");

System.out.println("Declaring Static Display function above by Interface obj arg ");

display(d);

display(e);

display(s);

}

}

==============================================================================

output

