**NAME :- VAIBHAV SHAILESH BISEN**

**ROLL NO:-233230**

**PRN:-230341220217**

**\*\*\*ASSIGNMENT DAY 1 & 2\*\*\***

**Q1: Write program to test Hello World. :-**

**package** day1asssignement;

**public** **class** q1 {

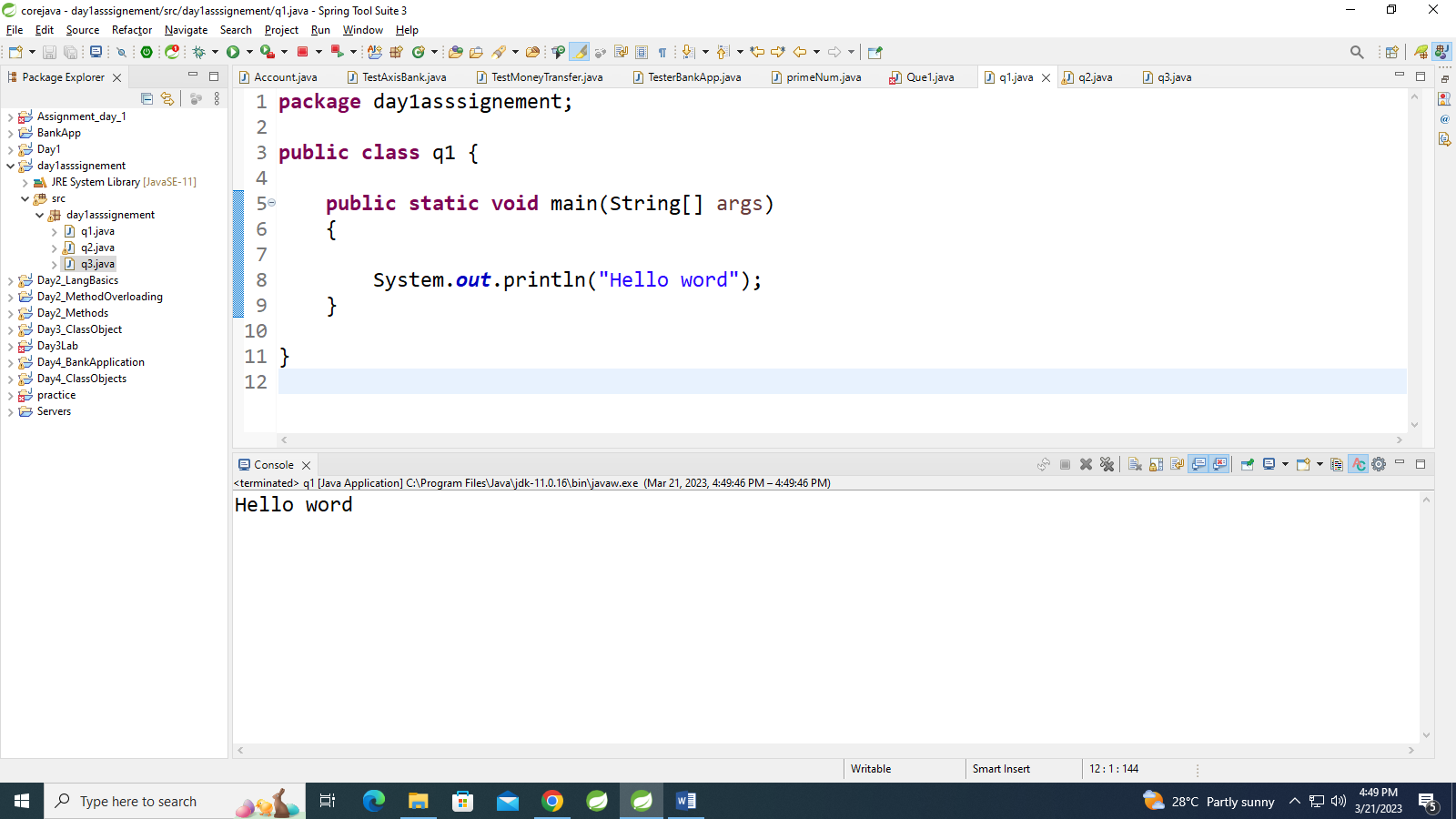
**public** **static** **void** main(String[] args)

{

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

}

}



**Q2:Write a program to addition of two numbers** .

**package** day1asssignement;

**import** java.util.Scanner;

**public** **class** q2 {

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

**int** a,b,c;

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

a=sc.nextInt();

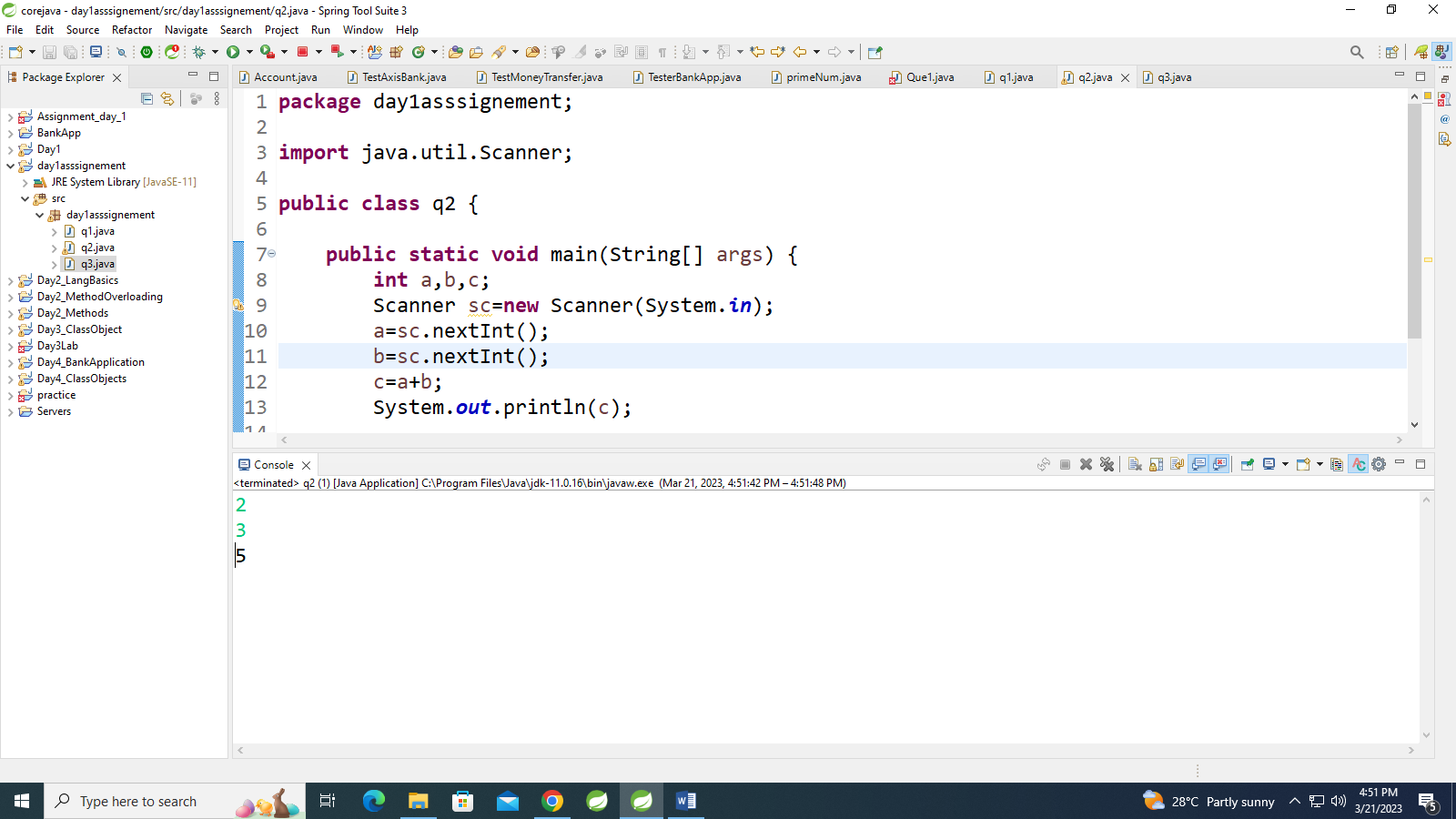
b=sc.nextInt();

c=a+b;

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

}

}



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

**package** day1asssignement;

**public** **class** q3 {

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

**int** a,b,c;

a=10;

b=20;

c=a;

a=b;

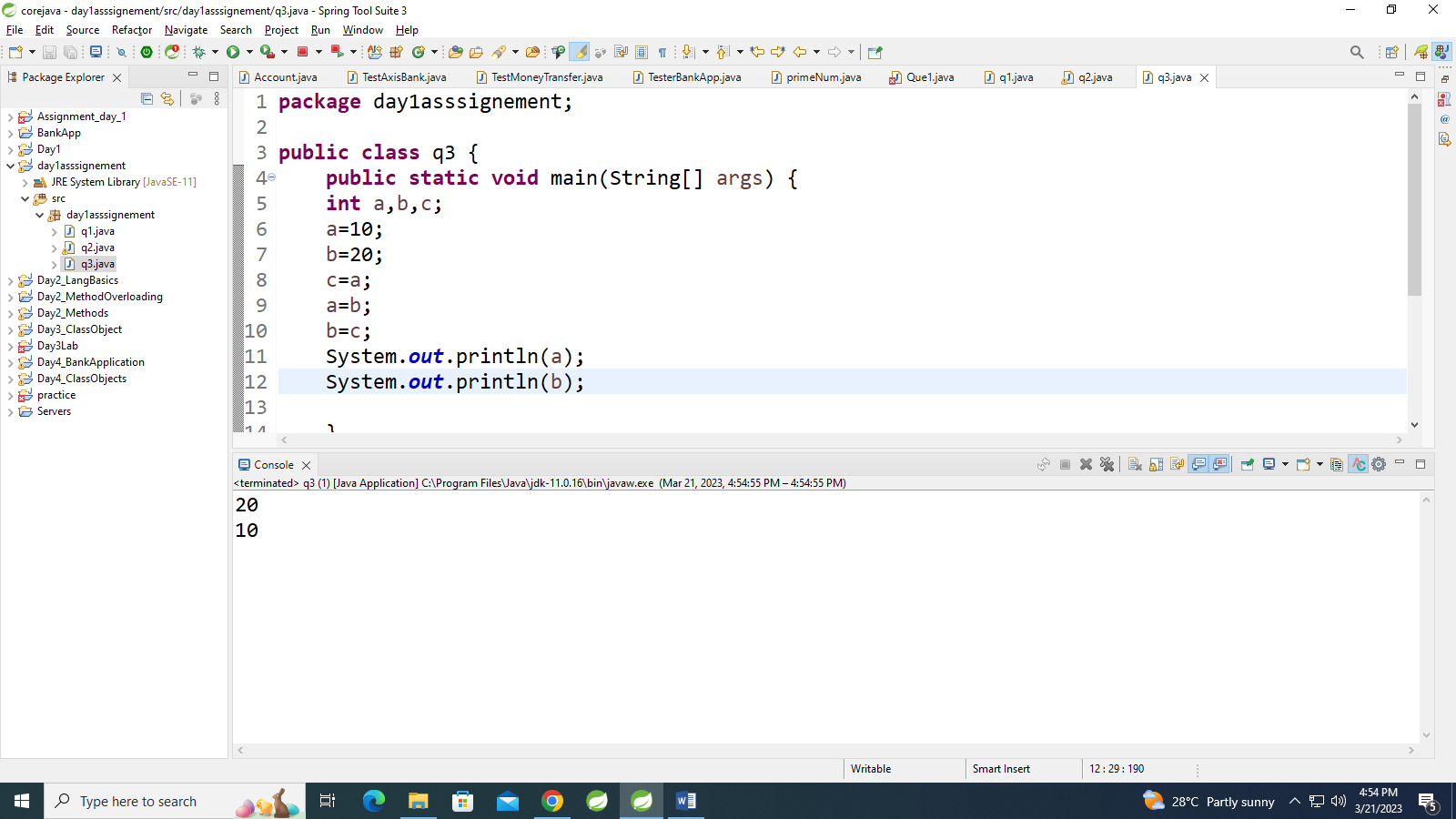
b=c;

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

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

}

}



.

Q4:Write a program to find factorial of a given number.

import java.util.Scanner;

public class factorial {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner c = new Scanner(System.in);

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

int n1;

n1 = c.nextInt();

int n2=1;

for( int i=1; i<=n1; i++) {

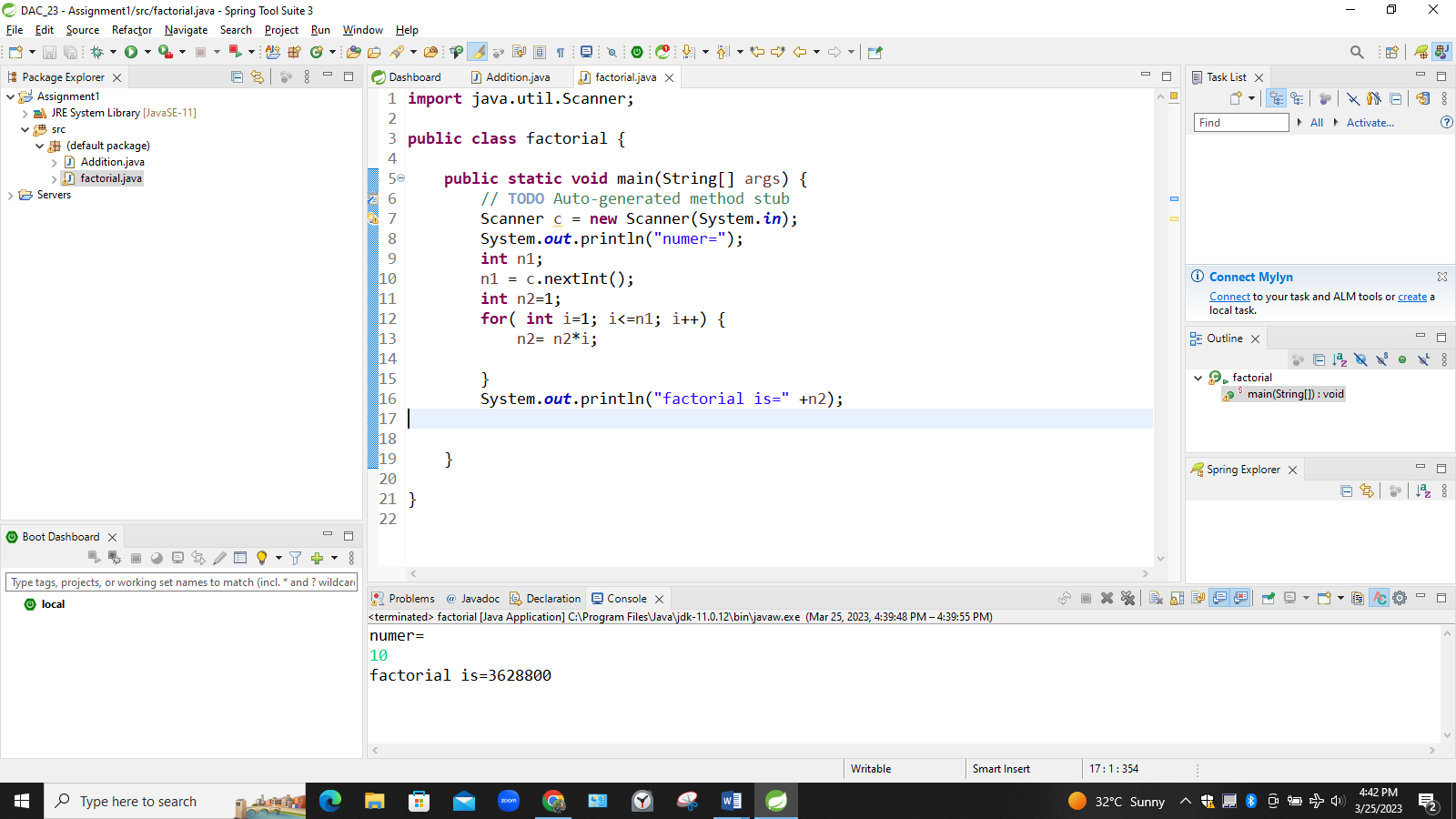
n2= n2\*i;

}

System.out.println("factorial is=" +n2);

}

}



**Q 5:Write a program to find m to the power n.**

**import** java.util.Scanner;

**public** **class** power {

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

// **TODO** Auto-generated method stub

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

System.***out***.println("first num is=");

**int** n1;

n1 = c.nextInt();

System.***out***.println("second num is=");

**int** n2;

n2= c.nextInt();

**int** n3=1;

**for**(**int** i=0; i<n2; i++) {

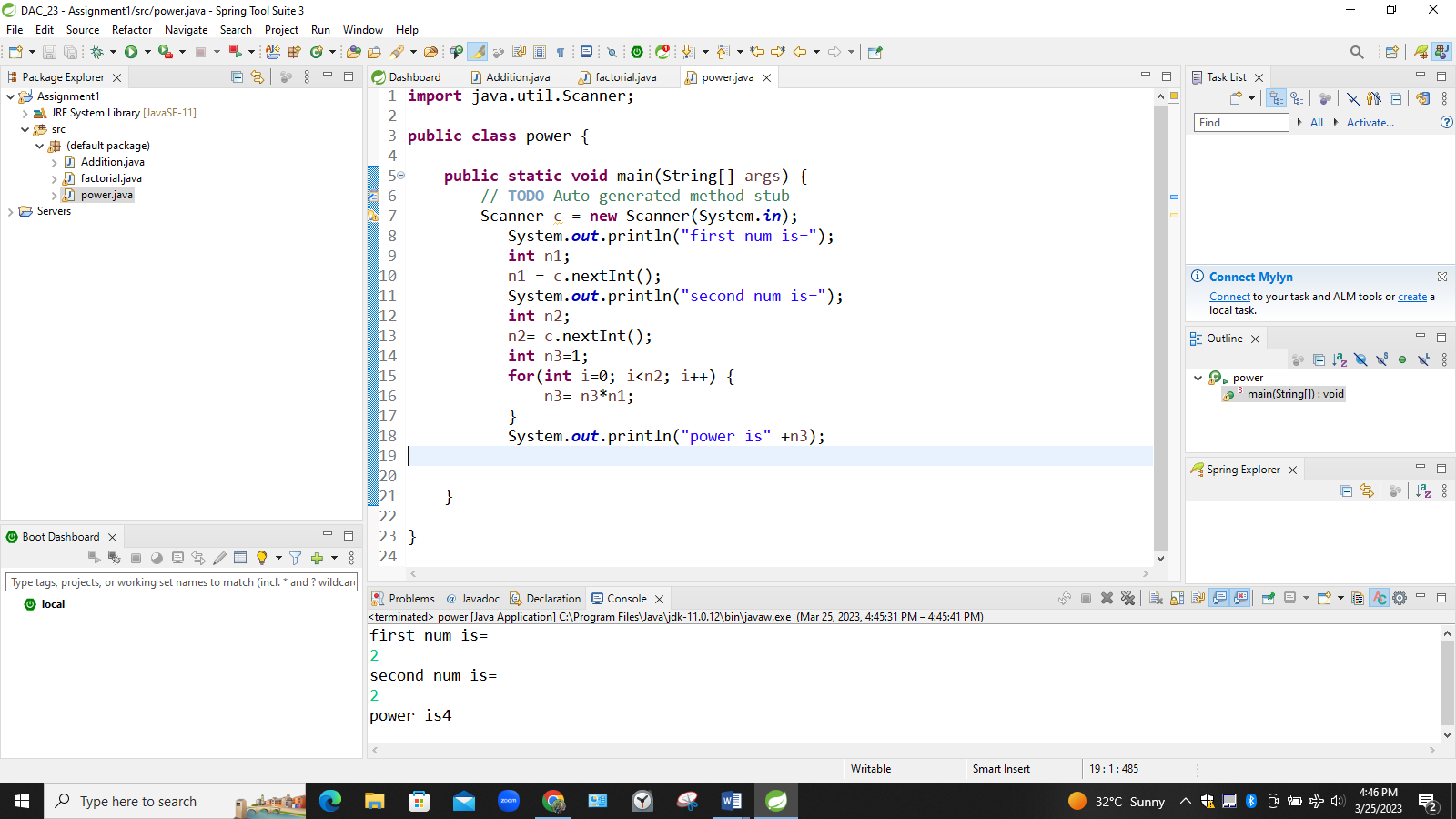
n3= n3\*n1;

}

System.***out***.println("power is" +n3);

}

}



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

**import** java.util.Scanner;

**public** **class** prime {

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

// **TODO** Auto-generated method stub

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

System.***out***.println("given number is");

**int** n;

n= c.nextInt();

**int** i, m=0, flag=0;

m= n/2;

**if**(n==0 || n==1) {

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

}**else** {

**for**(i=2; i<=m; i++) {

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

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

flag=1;

**break**;

}

}

**if** (flag==0) {

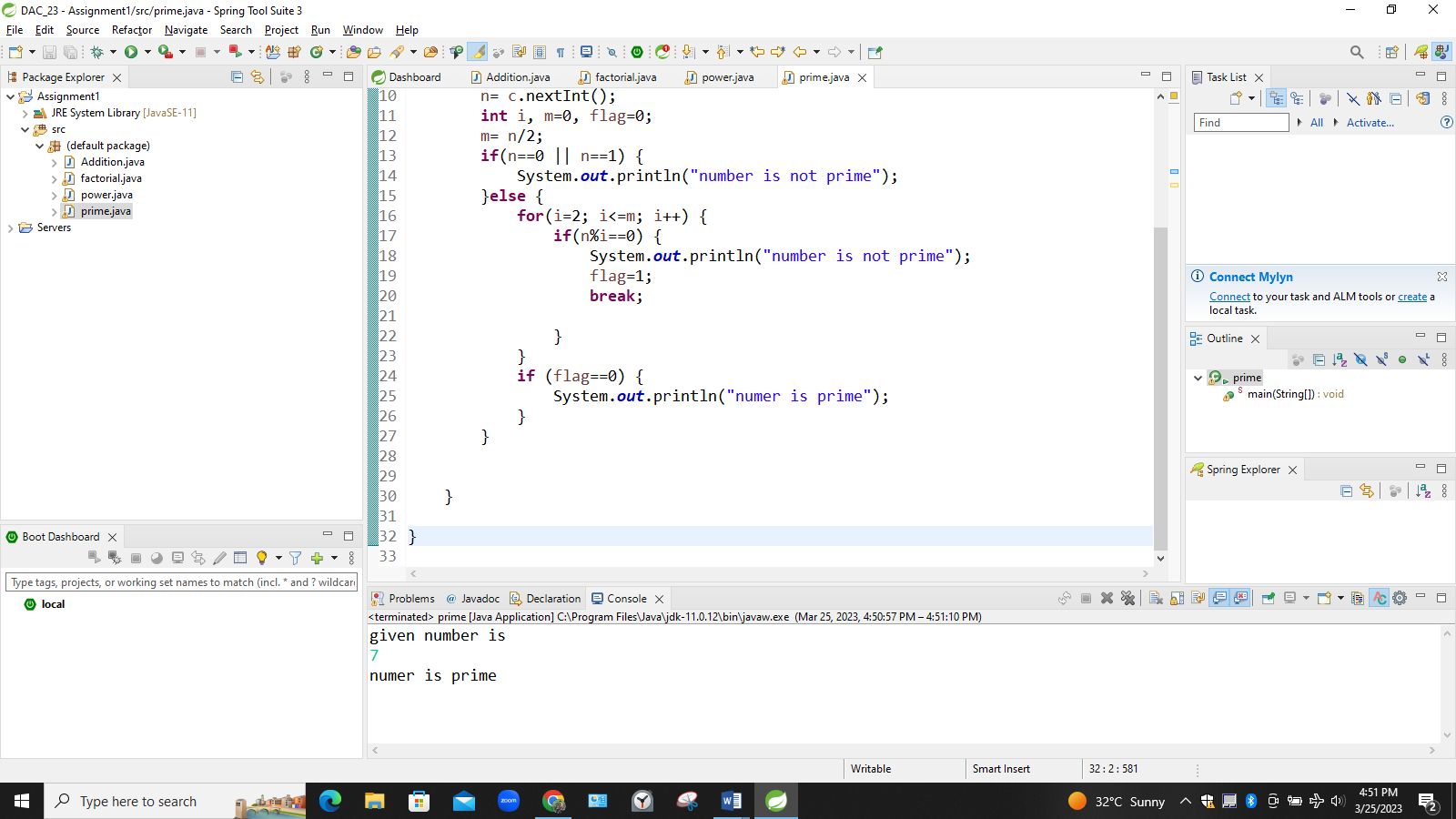
System.***out***.println("numer is prime");

}

}

}

}



**Q. 7:Sum of series :**

**1+2+3+….+n**

**import** java.util.Scanner;

**public** **class** sumseries {

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

// **TODO** Auto-generated method stub

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

System.***out***.println("given number");

**int** m;

m= c.nextInt();

**int** n=0;

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

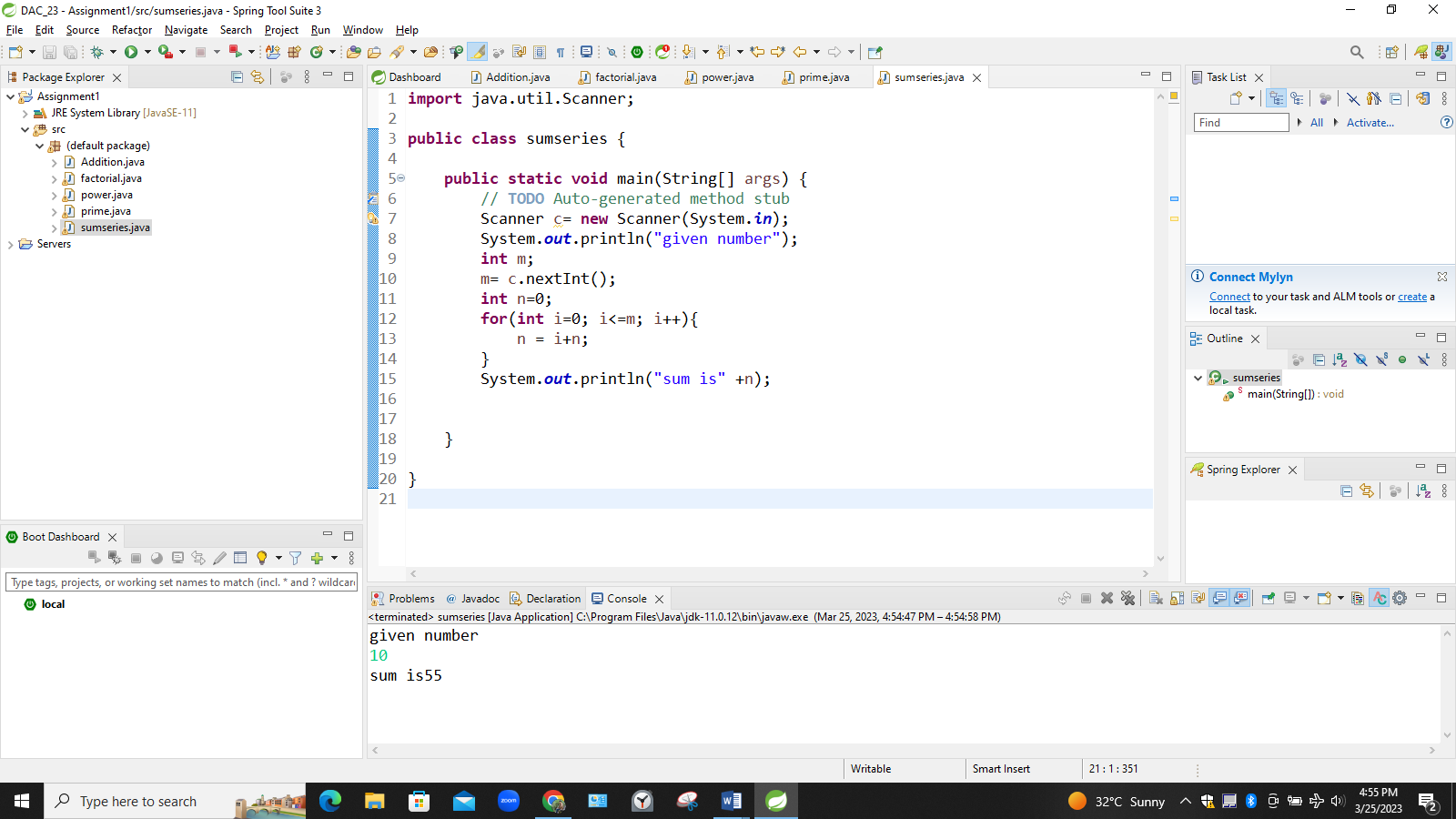
n = i+n;

}

System.***out***.println("sum is" +n);

}

}



**“ Q9. Write a program to find sum of all even and odd numbers between 1 to n.”**

**import** java.util.Scanner;

**public** **class** evenodd {

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

// **TODO** Auto-generated method stub

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

System.***out***.println("given number is");

**int** m;

m= c.nextInt();

**int** n=0;

**int** o=0;

**int** i;

**for**(i=0; i<=m; i++) {

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

n= n+i;

}**else** {

o=o+i;

}

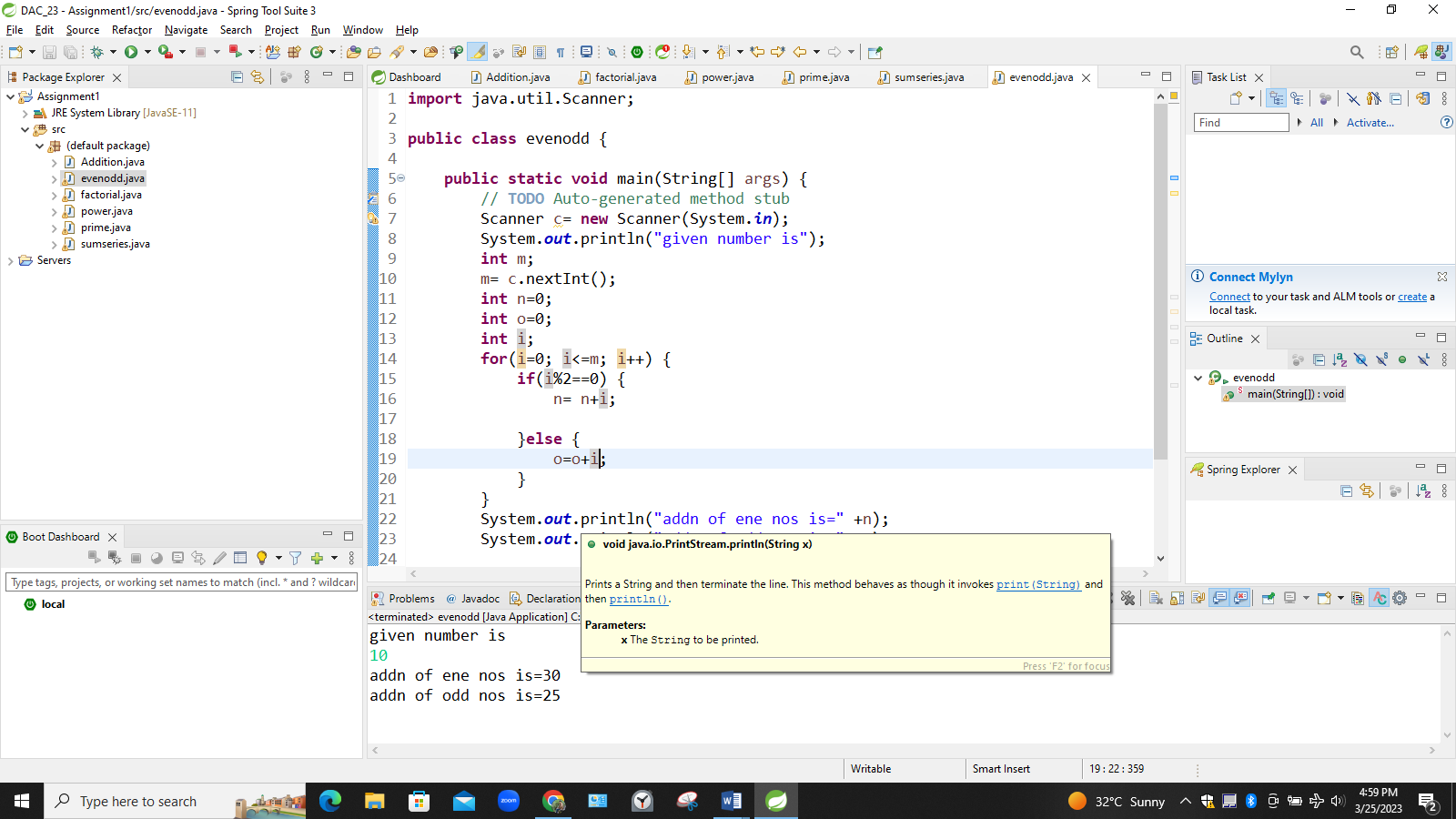
}

System.***out***.println("addn of ene nos is=" +n);

System.***out***.println("addn of odd nos is=" +o);

}

}



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

**import** java.util.Scanner;

**public** **class** reverse {

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

// **TODO** Auto-generated method stub

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

System.***out***.println("given number is");

**int** n;

n= c.nextInt();

**int** reverse=0;

**while**(n!=0) {

**int** remainder= n%10;

reverse= reverse\*10 + remainder;

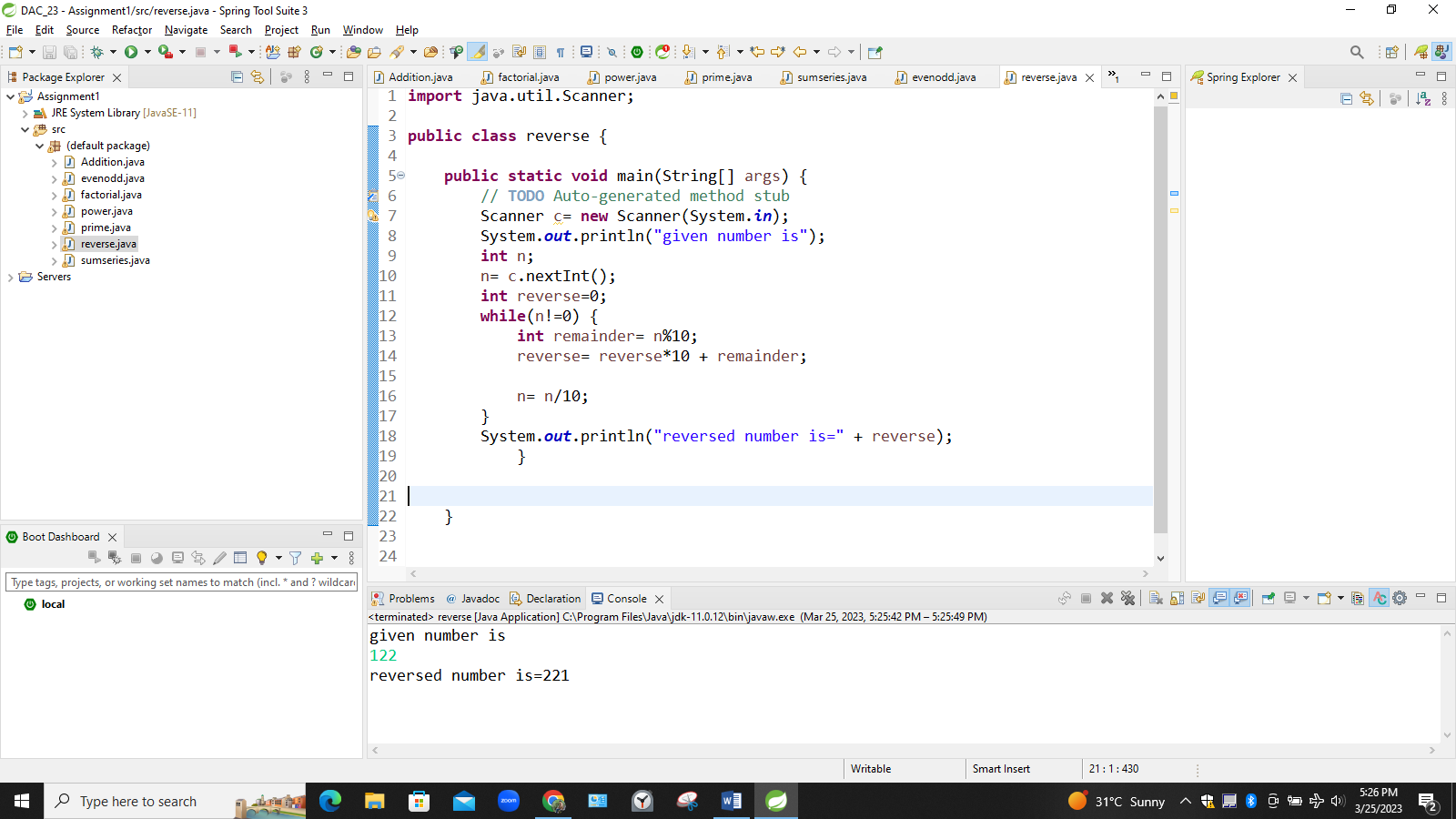
n= n/10;

}

System.***out***.println("reversed number is=" + reverse);

}

}



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

**import** java.util.Scanner;

**public** **class** primenumbers {

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

// **TODO** Auto-generated method stub

**int** i,n,counter, j;

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

System.***out***.print("Enter the n value : ");

n=scanner.nextInt();

**for**(j=2;j<=n;j++){

counter=0;

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

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

counter++;

}

}

**if**(counter==2)

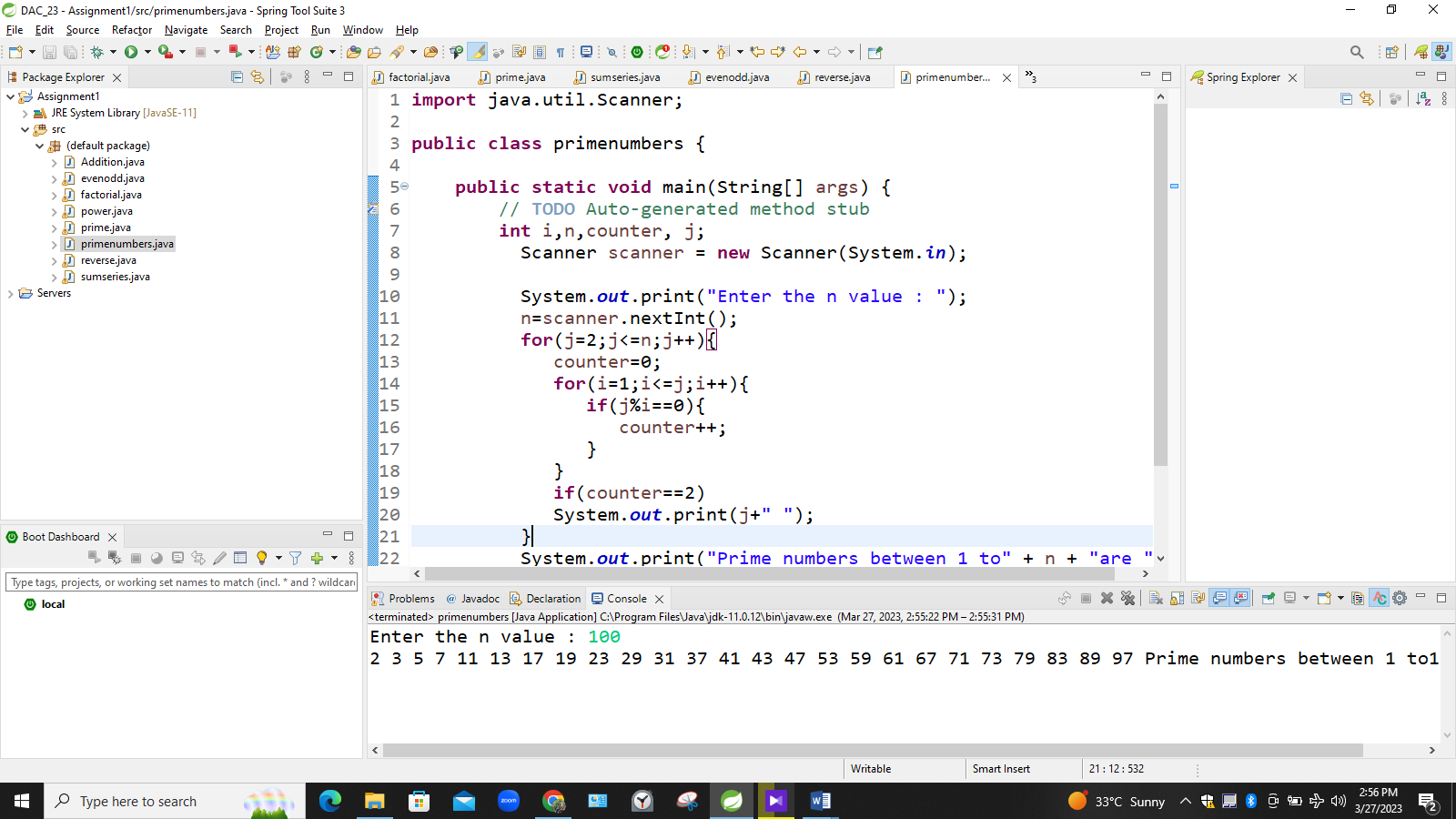
System.***out***.print(j+" ");

}

System.***out***.print("Prime numbers between 1 to" + n + "are ");

}

}



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

**import** java.util.Scanner;

**public** **class** armstrong {

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

// **TODO** Auto-generated method stub

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

System.***out***.println("first number is =");

**int** n1;

n1=c.nextInt();

System.***out***.println("second number is =");

**int** n2;

n2=c.nextInt();

System.***out***.println("third number is =");

**int** n3;

n3=c.nextInt();

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

System.***out***.println("first number is greater" +n1);

}**if**(n2>n1 && n2>n3) {

System.***out***.println("second number is greatest" +n2);

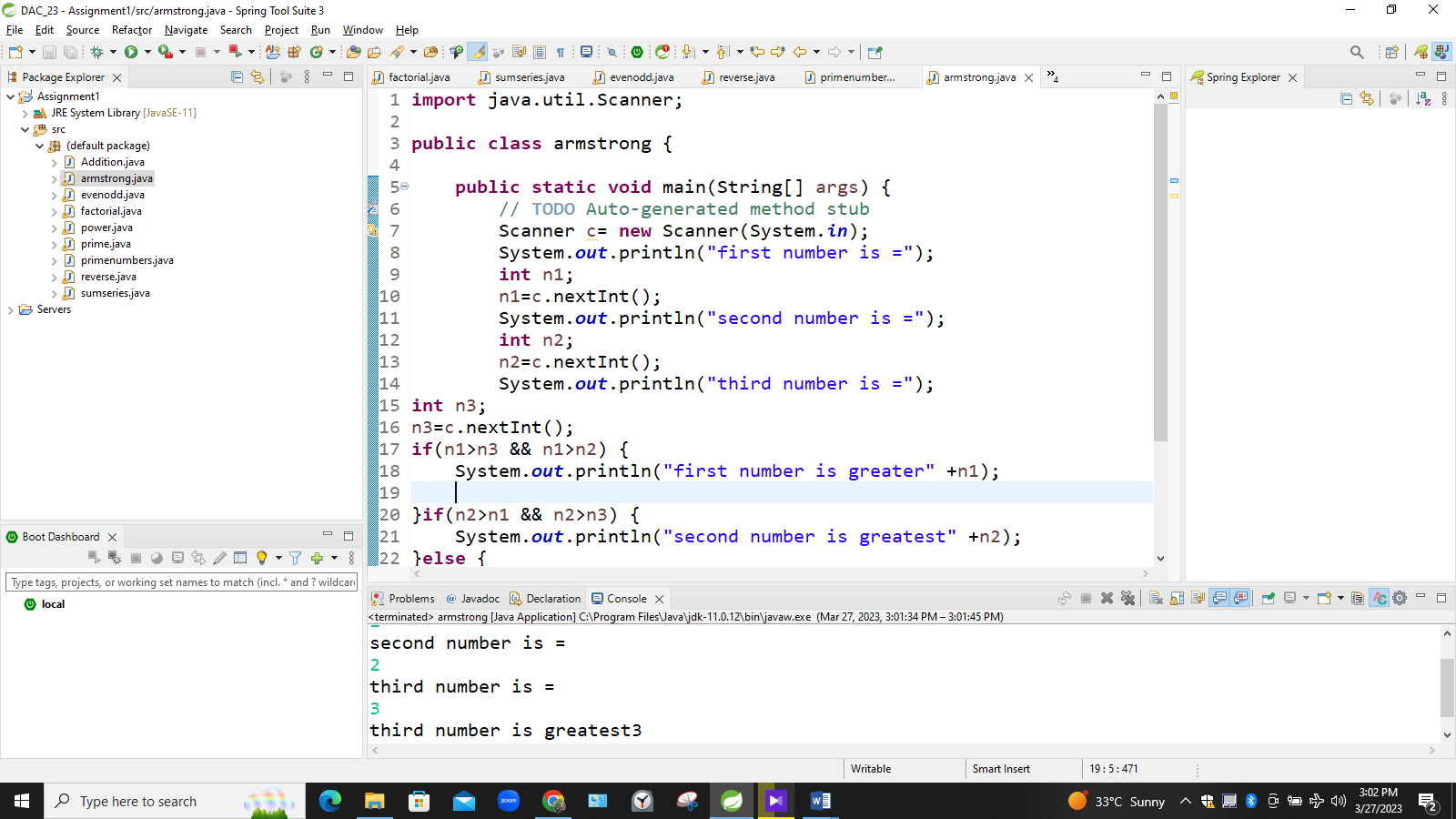
}**else** {

System.***out***.println("third number is greatest" +n3);

}

}

}



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

**import** java.util.Scanner;

**public** **class** Pizzamenu {

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

// **TODO** Auto-generated method stub

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

System.***out***.println("1: pizza 50rs 2:burger 40rs 3:coffee 30rs 4: total amount 5:exit");

**double** amount=0;

**int** qty, choice;

**do** {

System.***out***.println("enter your choice");

choice= c.nextInt();

**switch**(choice) {

**case** 1:

System.***out***.println("select quantity");

qty=c.nextInt();

amount += qty\*50;

**break**;

**case** 2:

System.***out***.println("select quantity");

qty=c.nextInt();

amount += qty\*40;

**break**;

**case** 3:

System.***out***.println("select quantity");

qty=c.nextInt();

amount += qty\*30;

**break**;

**case** 4:

System.***out***.println("total bill" +amount);

**break**;

**case** 5:

System.***out***.println("thank you visit again");

**break**;

**default**:System.***out***.println("choose correct option");

**break**;

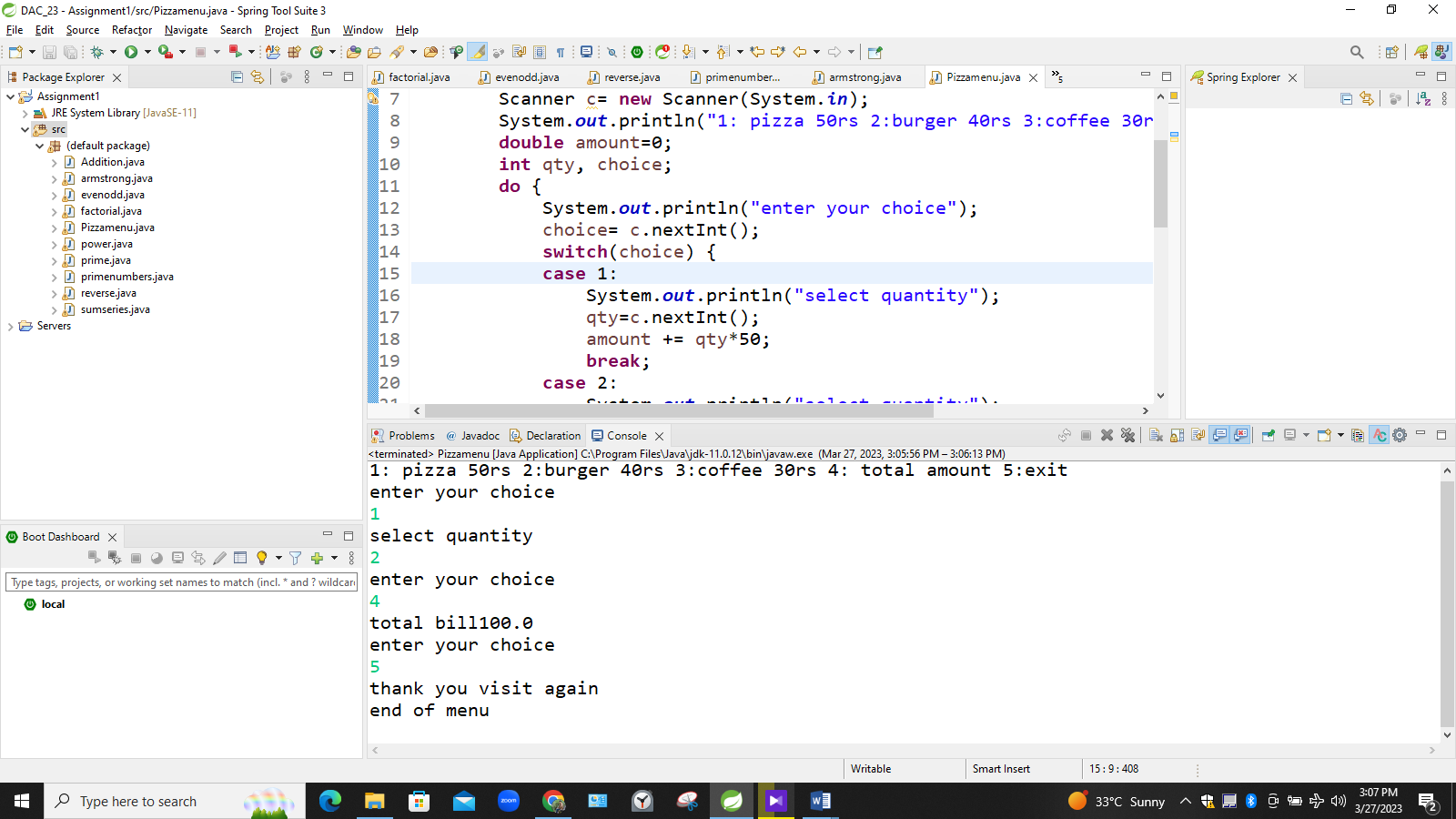
}

}**while**(choice!=5);

System.***out***.println("end of menu");

}

}



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

**import** java.util.Scanner;

**public** **class** arrayoperations {

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

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

**int**[] arr;

System.***out***.println("length of array");

**int** a= c.nextInt();

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

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

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

**int** b= c.nextInt();

arr[i]= b;

}

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

System.***out***.println("elements in array" +arr[j]);

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

**for**(**int** j=arr.length-1; j>=0; j-- ) {

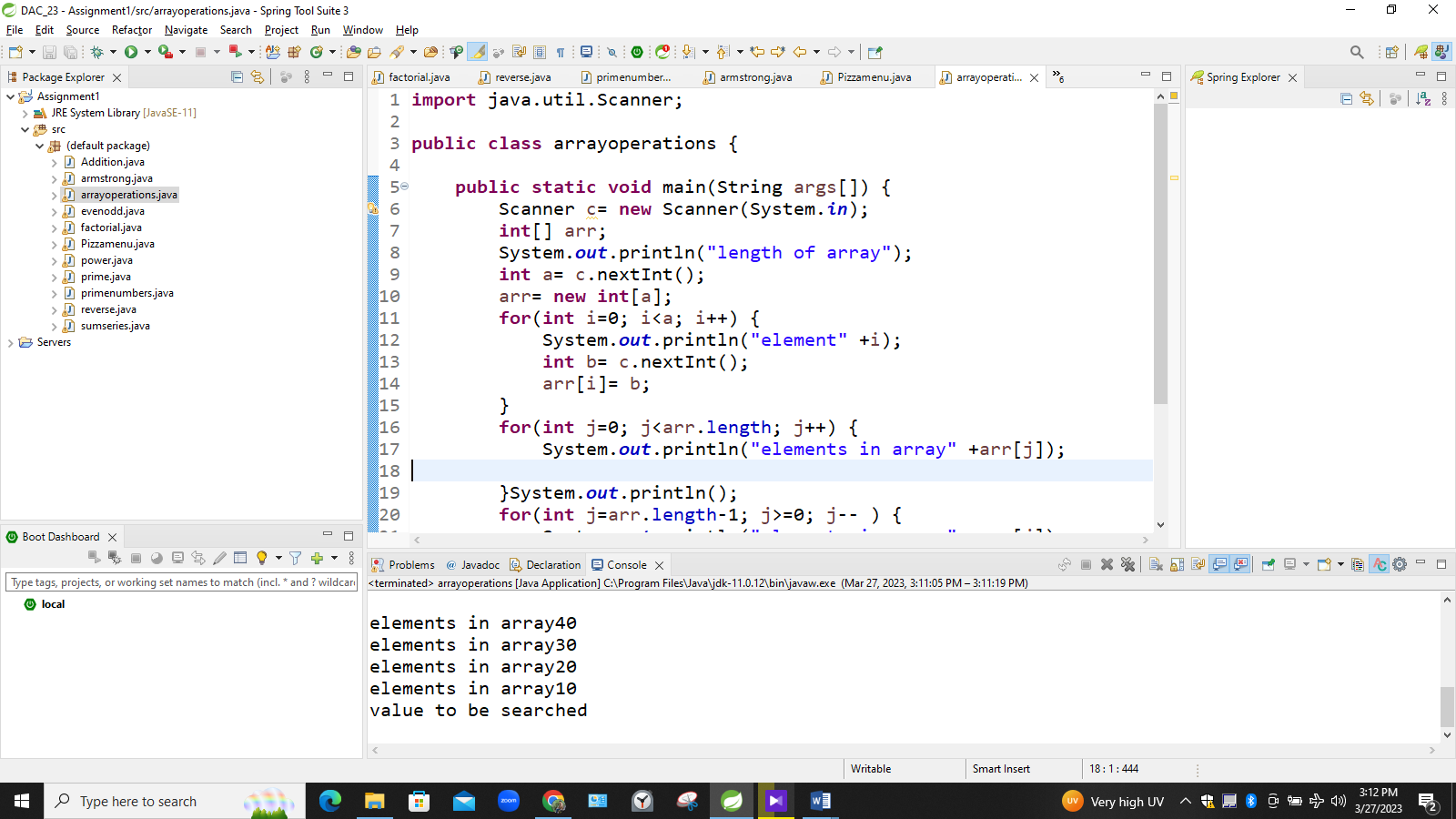
System.***out***.println("elements in array" +arr[j]);

}

System.***out***.println("value to be searched");

}

}



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

**import** java.util.Scanner;

**public** **class** arraysearch {

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

// **TODO** Auto-generated method stub

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

**int**[] arr;

System.***out***.println("length of array");

**int** a= c.nextInt();

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

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

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

**int** b= c.nextInt();

arr[i]= b;

}

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

System.***out***.println("elements in array" +arr[j]);

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

**for**(**int** j=arr.length-1; j>=0; j-- ) {

System.***out***.println("elements in array" +arr[j]);

}

System.***out***.println("value to be searched");

**int** z = c.nextInt();

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

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

**if**(arr[i]==z) {

flag=**true**;

}

}

**if**(flag) {

System.***out***.println("value is found");

}**else** {System.***out***.println("value not found");}

**int** sum=0;

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

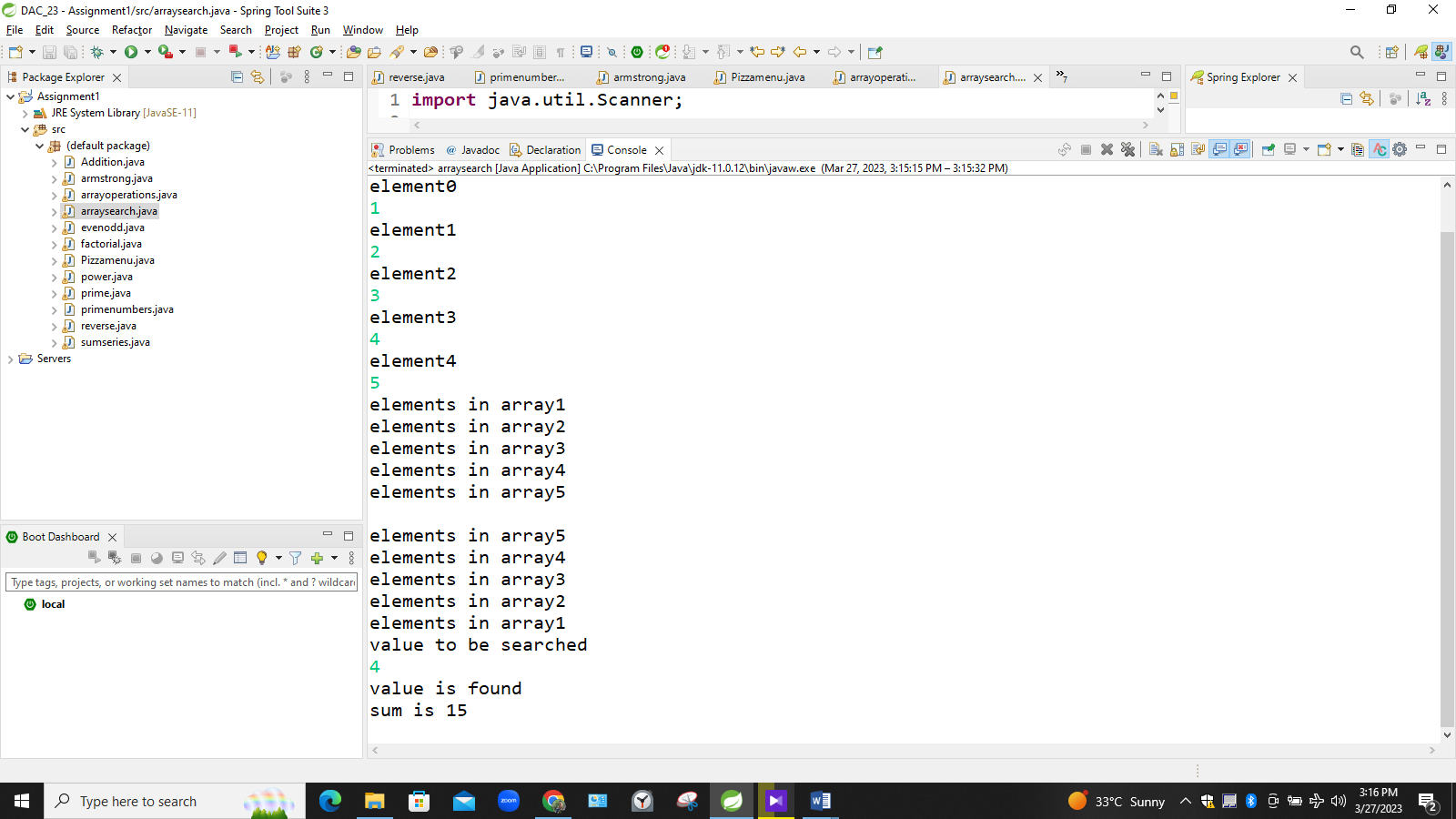
{ sum= arr[i]+sum;

}

System.***out***.println("sum is " +sum);

}

}



…….Thankyou……

**-: ASSIGNMENT\_DAY\_3&4;-**

**1:Create Date class with members day,month ,year.**

**Write no argument and parameterised constructor .**

**Create two object s and initialize them using no argument and parameterised constructor respectively.Print date using display function.**

**class** Date1 {

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

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

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

**public** Date1() {

**this**.day = 23;

**this**.month = 9;

**this**.year = 2023;

}

**public** Date1(**int** day, **int** month, **int** year) {

**this**.day = day;

**this**.month = month;

**this**.year = year;

}

// day, month, year

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

System.***out***.println("Day = " + day + "\nMonth = " + month + "\nYear = " + year);

}

}

**public** **class** Day3\_1 {

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

Date1 d1 = **new** Date1();

Date1 d2 = **new** Date1(29, 12, 1995);

d1.display();

d2.display();

}

**}**

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P :-**

**Day = 23**

**Month = 9**

**Year = 2023**

**Day = 29**

**Month = 12**

**Year = 1995**

**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

**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 = "Karan";

}

**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();

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

O/P :-

**ID = 43 Name = Karan 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.**

**class** Person {

**private** String name;

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

**private** String city;

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

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

**return** age;

}

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

**this**.age = age;

}

**public** String getCity() {

**return** city;

}

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

**this**.city = city;

}

**public** Person() {

**this**.name = "Abhishek";

**this**.age = 80;

**this**.city = "Pune";

}

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

**this**.name = name;

**this**.age = age;

**this**.city = city;

}

// name, age, city

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

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

}

}

**public** **class** Day3\_3 {

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

Person obj = **new** Person();

obj.display();

obj.setName("karan");

obj.setAge(95);

obj.setCity("Raipur");

obj.getName();

obj.getAge();

obj.getCity();

obj.display();

Person p2 = **new** Person("Heroine", 30, "Mumbai");

p2.display();

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P :-**

**Name = DHRUV**

**Age = 22**

**City = PORBANDAR**

**Name = KENIL**

**Age = 95**

**City = PUNE**

**Name = KATRINA**

**Age = 30**

**City = Mumbai**

**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

**4: Create a class Book with data members as bname,id,author,price. 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.**

**class** Book {

**private** String bname;

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

**private** String author;

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

**public** String getBname() {

**return** bname;

}

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

**this**.bname = bname;

}

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

**return** id;

}

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

**this**.id = id;

}

**public** String getAuthor() {

**return** author;

}

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

**this**.author = author;

}

**public** **double** getPrice() {

**return** price;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

// bname,id,author,price.

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

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

}

}

**public** **class** Day3\_4 {

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

Book obj = **new** Book();

obj.display();

obj.setBname("play with c");

obj.setId(101);

obj.setAuthor("karan");

obj.setPrice(5000.00);

obj.getBname();

obj.getId();

obj.getPrice();

obj.getAuthor();

obj.getPrice();

obj.display();

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P:-**

**Book Name = null**

**Id =0**

**Author = null**

**Price =0.0**

**Book Name = JAVA COMPLETE REFRENCE**

**Id =101**

**Author = HERBLET SCHIELD**

**Price =500.0**

**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

**5. Create a class Point with data members as x,y. 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**

**class** Point {

**private** **int** x;

**private** **int** 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;

}

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

System.***out***.println("X = " + x + "\n Y =" + y);

}

}

**public** **class** Day3\_5 {

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

Point p1 = **new** Point();

p1.display();

p1.setX(80);

p1.setY(20);

System.***out***.println(p1.getX());

System.***out***.println(p1.getY());

p1.display();

}

}

**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

**O/P:-**

**X = 0**

**Y = 0**

**80**

**20**

**X = 80**

**Y = 20**

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**6. Create a class ComplexNumber with data members real, imaginary. 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.**

**class** ComplexNumber {

**private** **int** real;

**private** **int** imaginary;

**public** **int** getReal() {

**return** real;

}

**public** **void** setReal(**int** real) {

**this**.real = real;

}

**public** **int** getImaginary() {

**return** imaginary;

}

**public** **void** setImaginary(**int** imaginary) {

**this**.imaginary = imaginary;

}

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

System.***out***.println("Real = " + real + "\nImaginary =" + imaginary);

}

}

**public** **class** Day3\_6 {

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

ComplexNumber c1 = **new** ComplexNumber();

c1.display();

c1.setReal(20);

System.***out***.println(c1.getReal());

c1.display();

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P:-**

**O/p-**

**Real = 0**

**Imaginary =0**

**20**

**Real = 20**

**Imaginary =0**

**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

**-: ASSIGNMENT\_DAY\_5;-**

**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.**

**package** com.student;

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

**private** **int** rno;

**private** **int** m1;

**private** **int** m2;

**private** **int** m3;

**public** **int** getRno() {

**return** rno;

}

**public** **int** getM1() {

**return** m1;

}

**public** **void** setM1(**int** m1) {

**this**.m1 = m1;

}

**public** **int** getM2() {

**return** m2;

}

**public** **void** setM2(**int** m2) {

**this**.m2 = m2;

}

**public** **int** getM3() {

**return** m3;

}

**public** **void** setM3(**int** m3) {

**this**.m3 = m3;

}

// int rno, int m1, int m2, int m3

**public** **void** disp() {

System.out.println("rollno " + **this**.getRno() + "\nm1= " + **this**.getM1() + "\nm2= " + **this**.getM2() + "\nm3 = "

+ **this**.getM3());

**int** total = **this**.getM1() + **this**.getM2() + **this**.getM3();

System.out.println("\nTotal = " + total);

**double** per = total / 3;

System.out.println(" Your percatage is : " + per);

**if** (per > 80)

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

**else** **if** (per > 65)

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

**else**

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

}

**public** **void** acceptInfo(**int** rno, **int** m1, **int** m2, **int** m3) {

System.out.println("Enter Marks m1, m2, m3");

{

**this**.rno = rno;

**this**.m1 = m1;

**this**.m2 = m2;

**this**.m3 = m3;

}

}

}

**package** StudentTester;

**import** com.student.Student;

**public** **class** Tester {

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

Student s1 = **new** Student();

s1.disp();

s1.acceptInfo(233003, 55, 69, 81);

s1.disp();

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P:-**

**rollno 0**

**m1= 0**

**m2= 0**

**m3 = 0**

**Total = 0**

**Your percatage is : 0.0**

**Grade = C**

**Enter Marks m1, m2, m3**

**rollno 233003**

**m1= 55**

**m2= 69**

**m3 = 81**

**Total = 205**

**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

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

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

**private** String name, city;

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

**public** String getName() {

**return** name;

}

**public** Person() {

System.***out***.println("This is under default constructor");

}

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

**this**.name = name;

**this**.city = city;

**this**.age = age;

}

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

**this**.name = name;

}

**public** String getCity() {

**return** city;

}

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

**this**.city = city;

}

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

**return** age;

}

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

**this**.age = age;

}

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

System.***out***.println("name" + " " + **this**.name + " " + "city " + " " + **this**.city + " " + "age" + " " + **this**.age);

}

}

**import** java.util.Scanner;

**public** **class** Person\_tester {

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

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

System.***out***.println("enter the name city and age");

Person p = **new** Day5\_Person(sc.next(), sc.next(), sc.nextInt());

p.display();

System.***out***.println("enter the name u want to update");

p.setName(sc.next());

p.getName();

System.***out***.println("enter the city u want to update");

p.setCity(sc.next());

p.getCity();

System.***out***.println("enter the age u want to update");

p.setAge(sc.nextInt());

p.getAge();

p.display();

sc.close();

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P:-**

**enter the name city and age**

**Abhishek**

**Raigarh**

**25**

**This is under default constructor**

**name Abhishek city  Raigarh age 25**

**enter the name u want to update**

**Vishal**

**enter the city u want to update**

**bhilai**

**enter the age u want to update**

**22**

**name Vishal city  bhilai age 22**

**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

**Q.3Create Date class with members day,month ,year.**

**Write no argument and parameterised constructor .**

**Create two object s and initialize them using no argument and parameterised constructor respectively.**

**Print date using display function.**

**class** Date1 {

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

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

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

**public** Date1() {

**this**.day = 23;

**this**.month = 9;

**this**.year = 2023;

}

**public** Date1(**int** day, **int** month, **int** year) {

**this**.day = day;

**this**.month = month;

**this**.year = year;

}

// day, month, year

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

System.***out***.println("Day = " + day + "\nMonth = " + month + "\nYear = " + year);

}

}

**public** **class** Day5\_2 {

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

Date1 d1 = **new** Date1();

Date1 d2 = **new** Date1(29, 12, 1995);

d1.display();

d2.display();

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P:-**

**Day = 23**

**Month = 9**

**Year = 2023**

**Day = 29**

**Month = 12**

**Year = 1995**

**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

**Create a class Book with data members as bname,id,author,price.**

**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.**

**class** Book {

**private** String bname;

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

**private** String author;

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

**public** String getBname() {

**return** bname;

}

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

**this**.bname = bname;

}

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

**return** id;

}

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

**this**.id = id;

}

**public** String getAuthor() {

**return** author;

}

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

**this**.author = author;

}

**public** **double** getPrice() {

**return** price;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

// bname,id,author,price.

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

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

}

}

**public** **class** Day5\_3 {

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

Book obj = **new** Book();

obj.display();

obj.setBname("play with c");

obj.setId(101);

obj.setAuthor("karan");

obj.setPrice(5000.00);

obj.getBname();

obj.getId();

obj.getPrice();

obj.getAuthor();

obj.getPrice();

obj.display();

}

}

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

O/P:-

Book Name = null

Id =0

Author = null

Price =0.0

Book Name = play with c

Id =101

Author = karan

Price =5000.0

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

DAY5\_6

**6:create BankAccount aaplication for operations like withdraw ,deposite and moneyTransfer.**

**Create menu drive program for bank operations..**

**package** banking;

**import** java.util.Scanner;

**public** **class** Ac

{

**private** **int** acid;

**private** String name, email, city;

**private** **double** bal;

**public** Ac(**int** acid, String name, String email, String city, **double** bal) {

**this**.acid = acid;

**this**.name = name;

**this**.email = email;

**this**.city = city;

**this**.bal = bal;

}

**public** **int** getAcid() {

**return** acid;

}

**public** **void** setAcid(**int** acid) {

**this**.acid = acid;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getCity() {

**return** city;

}

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

**this**.city = city;

}

**public** **double** getBal() {

**return** bal;

}

**public** **void** setBal(**double** bal) {

**this**.bal = bal;

}

**public** **void** deposit(**double** amt) {

**this**.bal+=amt;

System.***out***.println("Avail Bal = "+bal);

}

**public** **void** withdraw(**double** amt) {

**if**(bal>amt) {

System.***out***.println("Enter amount to be withdrawn");

**this**.bal-=amt;

System.***out***.println("Avail Bal = "+bal);

}**else**

System.***out***.println("Insufficent balance");

}

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

**public** **void** moneyTransfer(Ac rec, **double** amt) {

**this**.bal -= amt;

rec.bal += amt;

System.***out***.println("Money transfered from "+**this**.getName()+" to "+rec.getName() );

System.***out***.println(**this**.getName()+" ---- "+**this**.getBal()+"\n"+rec.getName()+" --- "+rec.getBal());

}

**public** **void** disp() {

//int acid, String name, String email, String city, double bal

System.***out***.println("Details of holder are:\n"+"ID = "+**this**.acid+", Name = "+**this**.getName()+", String email = "+**this**.getEmail()+", City = "+**this**.getCity()+", Balance = "+**this**.getBal());

}

}--------------------------------------------

**package** tester;

**import** java.util.\*;

**import** banking.Ac;

**public** **class** TesterBank {

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

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

// Ac(int acid, String name, String email, String city, double bal)

Ac a1 = **new** Ac(43, "Karan", "karan29@gmail.com", "Raipur", 25000);

// Ac a2;

// System.out.println("Enter Ac 2 detsils, id- name- email- city- avail bal");

// a2 = new Ac(sc.nextInt(),sc.next(), sc.next(), sc.next(), sc.nextDouble());

Ac a3 = **new** Ac(25, "Bhavesh", "bhavi@gmail.com", "Raipur", 20000);

a1.disp();

a3.disp();

System.***out***.println("Enter amount to be transfered");

**double** amt = sc.nextDouble();

a1.moneyTransfer(a3, amt);

sc.close();

}

}

-------------------------------------------

**package** tester;

**import** java.util.\*;

**import** banking.Ac;

**public** **class** TesterAcHo {

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

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

System.***out***.println("Enter Ac Holder detsils, id- name- email- city- bal");

Ac a4;

a4 = **new** Ac(sc.nextInt(), sc.next(), sc.next(), sc.next(), sc.nextDouble());

a4.disp();

Ac a5 = **null**;

**int** choice = 0;

**do** {

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

System.***out***.println(

"1.Create Ac\n2.Show Ac\n3.Deposit\n4.Withdraw\n5.CheckBal\n6.Update Email\n7.Update City\n8.Money Transfer\n9.Exit");

choice = sc.nextInt();

**switch** (choice) {

**case** 1: // Create Ac

System.***out***.println("Enter Id");

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

**if** (a4.getAcid() != id) {

a5 = **new** Ac(sc.nextInt(), sc.next(), sc.next(), sc.next(), sc.nextDouble());

System.***out***.println("Holder with " + a5.getName() + " has been created.");

}

**break**;

**case** 2: // Show Ac

**if** (a5 != **null**)

a5.disp();

**break**;

**case** 3: // Deposit

**if** (a5 != **null**) {

a5.deposit(sc.nextDouble());

a5.getBal();

}

**break**;

**case** 4: // Withdraw

**if** (a5 != **null** && a5.getBal() > sc.nextDouble())

a5.withdraw(sc.nextDouble());

a5.getBal();

**break**;

**case** 5: // Check Bal

**if** (a5 != **null**)

System.***out***.println(a5.getBal());

**break**;

**case** 6: // Update email

**if** (a5 != **null**) {

System.***out***.println("Old email = " + a5.getEmail());

System.***out***.println("Enter new Email");

a5.setEmail(sc.next());

System.***out***.println("New Email is : " + a5.getEmail());

}

**break**;

**case** 7: // Update city

**if** (a5 != **null**) {

System.***out***.println("Old city = " + a5.getCity());

System.***out***.println("Enter new City");

a5.setCity(sc.next());

System.***out***.println("New Email is : " + a5.getCity());

}

**break**;

**case** 8: // Money Transfer

System.***out***.println("Enter amount to be transfered");

**double** amt = sc.nextDouble();

Ac sen, rec = **null**;

System.***out***.println("Enetr sender & reciever id");

**int** id1 = sc.nextInt(), id2 = sc.nextInt();

// boolean idmatch = false;

**if** (a5.getAcid() == id1 && a4.getAcid() == id2) {

sen = a5;

rec = a4;

sen.moneyTransfer(rec, amt);

}

**break**;

**default**:

System.***out***.println("Invalid choice, Please try again later");

**break**;

}

} **while** (choice < 9);

sc.close();

}

}

---------------------------------------------------

**package** tester;

**import** java.util.\*;

**import** banking.Ac;

**public** **class** AcArray {

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

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

Ac hld[] = **new** Ac[4];

**int** index = 0;

**int** y = 0;

// int acid, String name, String email, String city, double bal

hld[0] = **new** Ac(1, "Ka", "k@gmail.com", "Raipur", 20000);

hld[1] = **new** Ac(2, "Ab", "a@gmail.com", "Raigar", 25000);

hld[2] = **new** Ac(3, "Bh", "b@gmail.com", "Pune", 30000);

hld[3] = **new** Ac(4, "Sid", "sid@gmail.com", "Bhilai", 35000);

**for** (Ac x : hld)

index++;

**int** choice = 0;

**do** {

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

System.***out***.println("1.Create Ac\n2.Show Ac\n3.Deposit\n4.Withdraw\n5.CheckBal\n6.Money Transfer\n7.Exit");

choice = sc.nextInt();

**switch** (choice) {

**case** 1: // Create Ac

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

**if** (hld[i] == **null**) {

System.***out***.println("Array is empty \n Enter Details");

hld[i] = **new** Ac(sc.nextInt(), sc.next(), sc.next(), sc.next(), sc.nextDouble());

y++;

System.***out***.println("New Account has been created");

} **else** **if** (hld[i] != **null** && i < hld.length) {

System.***out***.println("Enter new ac details");

hld[i] = **new** Ac(sc.nextInt(), sc.next(), sc.next(), sc.next(), sc.nextDouble());

y++;

System.***out***.println("New Account has been created");

} **else**

System.***out***.println("New Account can not be created and array is full");

}

**break**;

**case** 2: // show all A/C

System.***out***.println("All Existing ac holders");

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

hld[i].disp();

**break**;

**case** 3: // Deposit

System.***out***.println("Enter id in which amt to be deposit");

**int** x\_id = sc.nextInt();

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

**if** (hld[i].getAcid() == x\_id) {

System.***out***.println("Enter amount to be deposited");

hld[i].deposit(sc.nextDouble());

}

}

**break**;

**case** 4: //withdraw

System.***out***.println("Enter id in which amt to be withdraw"); **int**

x\_id=sc.nextInt(); **for** (**int** i=0; i<index; i++) { **if**(hld[i].getAcid()==x\_id) {

System.***out***.println("Enter amount to be deposited");

hld[i].deposit(sc.nextDouble()); } } **break**;

**case** 5: // bal check

System.***out***.println("Enter id whose ac details to be checked");

**int** z\_id = sc.nextInt();

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

**if** (hld[i].getAcid() == z\_id)

hld[i].getBal();

}

**break**;

**case** 6:

System.***out***.println("display all ac details");

Ac sen = **null**;

Ac rec = **null**;

**boolean** present = **false**;

System.***out***.println("Enter k1 and k2 ids");

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

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

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

**if** (sid == hld[i].getAcid())

sen = hld[i];

present = **true**;

}

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

**if** (rid == hld[i].getAcid())

rec = hld[i];

present = **true**;

}

**if** (present == **true**) {

System.***out***.println("Enter amount to be transfered");

**double** amt = sc.nextDouble();

sen.moneyTransfer(rec, amt);

}

**default**:

System.***out***.println("Invalid choice, Please try again later");

**break**;

}

} **while** (choice < 9);

sc.close();

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P:-**

**Enter choice 1.**

**Create Ac 2.**

**Show Ac 3.D eposit 4. Withdraw 5. CheckBal 6. Money Transfer 7.Ex it 2**

**All Existing**

**ac holders**

**Details of**

**holder are:ID=1,Name=Ka,**

**String email = k@gmail.com, City = Raipur, Balance = 20000.0**

**Details of**

**holder are:ID=2,Name=Ab,**

**String email = a@gmail.com, City = Raigar, Balance = 25000.0**

**Details of**

**holder are:ID=3,Name=Bh,**

**String email = b@gmail.com, City = Pune, Balance = 30000.0**

**Details of**

**holder are:ID=4,Name=Sid,**

**String email = sid@gmail.com, City = Bhilai, Balance = 35000.0**

**Enter choice 1.**

**Create Ac 2.**

**Show Ac 3.D eposit 4. Withdraw 5. CheckBal 6. Money Transfer 7.Ex it 3**

**Enter id**

**in which**

**amt to**

**be deposit 2**

**Enter amount**

**to be deposited 10000**

**Avail Bal = 35000.0**

**Enter choice 1.**

**Create Ac 2.Show Ac 3.D eposit 4. Withdraw 5. CheckBal 6. Money Transfer 7.Ex it 6**

**display all**

**ac details**

**Enter k1**

**and k2 ids 2 3**

**Enter amount**

**to be transfered 10000**

**Money transfered**

**from Ab**

**to Bh Ab----25000.0 Bh---40000.0**

**Invalid choice, Please try**

**again later**

**Enter choice 1.**

**Create Ac 2.Show Ac 3.D eposit 4. Withdraw 5. CheckBal 6. Money Transfer 7.Ex**

**It**

**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

**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:Create diff package and add class inside that.**

**Try to access one package class in another package....**

**(chk default access specifier)**

**package** day5\_logic;

**public** **class** ArrayStud {

**int** rollNo;

**private** String name;

**private** String adress;

**public** **static** **int** autoIn;

**public** ArrayStud(String name, String adress) {

autoIn++;//to increment roll number automatically

**this**.rollNo=autoIn;

**this**.name = name;

**this**.adress = adress;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** String getAdress() {

**return** adress;

}

**public** **void** setAdress(String adress) {

**this**.adress = adress;

}

**public** **int** getRollNo() {

**return** rollNo;

}

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

{

System.out.println("Roll\_No:"+rollNo+" Name:"+name+" Adress:"+adress);

}

}

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

**package** A7tester;

**import** java.util.Scanner;

**import** day5\_logic.ArrayStud;

**public** **class** TestStud {

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

ArrayStud s1=**new** ArrayStud("rutvik","surat");

s1.display();

ArrayStud s2=**new** ArrayStud("toufik","latur");

s2.display();

//System.out.println(s2.rollNo);//------>checked package level visibility//error

ArrayStud st[] =**new** ArrayStud[5];

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

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

{

System.out.println("Enter the name and city for a student "+(i+1));

st[i]=**new** ArrayStud(sc.next(),sc.next());

}

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

{

System.out.println(st[i].getName());

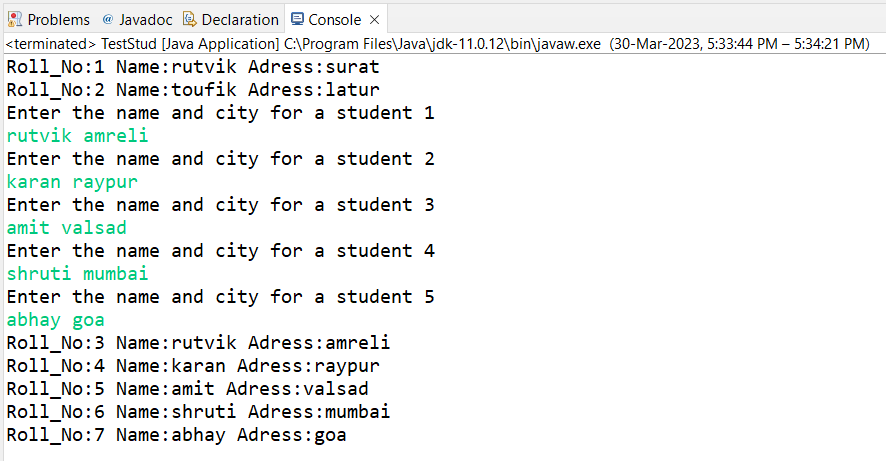
}

}

}

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

O/P:-



<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

day6\_1

FOR PRACTICE ONLY

**package** com.logic;

**public** **class** Apple **extends** Fruit

{

**public** Apple(String color, **float** wt) {

**super**(color, wt);

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "Apple [color=" + color + ", toString()=" + **super**.toString() + "]";

}

**public** **void** appy()

{

System.out.println("Appy is an apple extracts");

}

**public** **void** taste()

{

System.out.println("Appy fiz is awsome");

}

}

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

**package** com.logic;

**public** **class** Banana **extends** Fruit

{

**public** Banana(String color, **float** wt) {

**super**(color, wt);

}

@Override

**public** String toString() {

**return** "Banana [color=" + color + "toSting="+ **super**.toString() +"]";

}

**public** **void** shake() {

System.out.println(" By banana milkshake is being made");

}

**public** **void** taste()

{

System.out.println("Banana shake makes the brain awake");

}

}

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

**package** com.logic;

**public** **class** Mango **extends** Fruit

{

**public** Mango(String color, **float** wt) {

**super**(color, wt);

}

@Override

**public** String toString() {

**return** "Mango [color=" + color + ", toString()=" + **super**.toString() + "]";

}

**public** **void** taste()

{

System.out.println("With mango juice and shake both can be made");

}

}

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

**package** com.testLogic;

**import** com.logic.Apple;

**import** com.logic.\*;

**public** **class** TestFruit {

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

Fruit f1=**new** Fruit ("multi",0.555f);

System.out.println(f1);

f1.taste();

Apple a1=**new** Apple ("red",0.985f);

System.out.println(a1);

a1.taste();

Banana b1=**new** Banana ("yellow",2.200f);

System.out.println(b1);

b1.taste();

Mango m1=**new** Mango ("orange",0.780f);

System.out.println(m1);

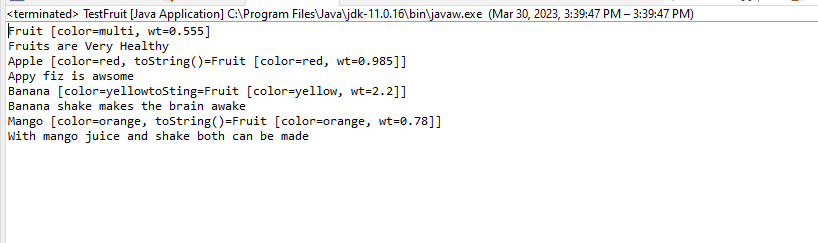
m1.taste();

}

}

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

O/P:-



<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

DAY6\_2

**package** com.logic;

**class** Date

{

**private** **int** dd;

**private** **int** mm;

**private** **int** yy;

**public** Date(**int** dd, **int** mm, **int** yy)

{

**this**.dd = dd;

**this**.mm = mm;

**this**.yy = yy;

}

**public** Date() {

**this**.dd = 12;

**this**.mm = 12;

**this**.yy = 2000;

}

@Override

**public** String toString() {

**return** "Date [dd=" + dd + ", mm=" + mm + ", yy=" + yy + "]";

}

}

**public** **class** Person **extends** Date

{

**private** String name;

@Override

**public** String toString() {

**return** "Person [name=" + name + ", toString()=" + **super**.toString() + "]";

}

**public** Person(**int** dd, **int** mm, **int** yy, String name) {

**super**(dd, mm, yy);

**this**.name = name;

}

**private** **void** helper() { System.out.println("INSIDE person private helper method");}

**public** **void** testDisplay()

{

System.out.println("in side Person class :");

helper();

}

}

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

**package** com.logic;

**public** **class** Emp **extends** Person {

**private** **int** empid;

**private** **double** sal;

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

**static** {

count = 1000;

}

**public** Emp(**int** dd, **int** mm, **int** yy, String name, **double** sal) {

**super**(dd, mm, yy, name);

**this**.sal = sal;

**this**.empid = ++count;

}

@Override

**public** String toString() {

**return** "Emp [empid=" + empid + ", sal=" + sal + ", toString()=" + **super**.toString() + "]";

}

**public** **void** testDisplay() {

System.out.println("in side EMP class :");

}

}

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

//super(dd, mm, yy, name, empid, sal);

// this.dept\_name = dept\_name;

// this.team\_size = team\_size;

**package** com.testLogic;

//import com.logic.\*;

**import** com.logic.Emp;

**import** com.logic.Manager;

**import** com.logic.Person;

**public** **class** TestPEM {

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

Person p1 = **new** Person(1,2,2000,"rutvik");

p1.testDisplay();

System.out.println();

System.out.println(p1);

Emp e1 = **new** Emp(15,1,2001,"karan",100000);

e1.testDisplay();//method override

System.out.println();

System.out.println(e1);

Manager m1 = **new** Manager(14,10,1901,"abhay",100000,"FrontEnd",50);

m1.testDisplay();//method override

System.out.println();

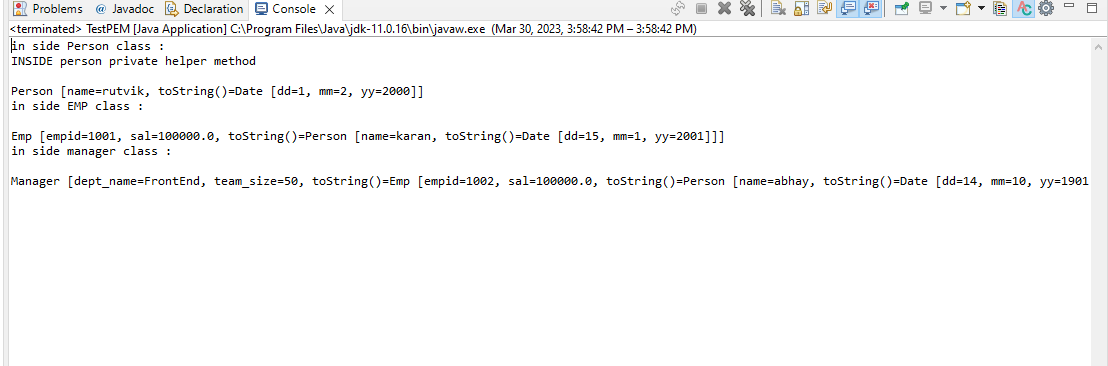
System.out.println(m1);

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P:-**



**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

DAY 7

day7\_1

/\*

\* 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]);

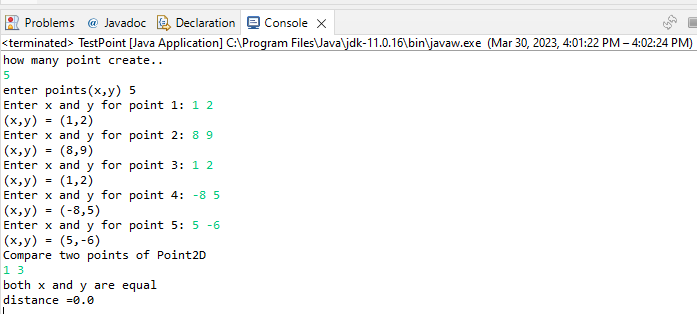
System.***out***.println("distance ="+res);

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P:-**



**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

day7\_2

Task 2

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;

}

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

@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);

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P:-**

**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**

**visit again !!!**

**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

day7\_3

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");

}

}

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

**O/P:-**

**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

**DAY 8**

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

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

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

**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** hight) {

**super**(x, y);

**this**.width=width;

**this**.height=hight;

}

**public** **double** getWidth() {

**return** width;

}

**public** **void** setWidth(**double** width) {

**this**.width = width;

}

**public** **double** getHeight() {

**return** height;

}

**public** **void** setHeight(**double** hight) {

**this**.height = hight;

}

@Override

**public** String toString() {

**return** **super**.toString() +"Rectangle [width=" + width + ", hight=" + 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 ,redius");

System.out.println("Enetr 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 ,redius");

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);

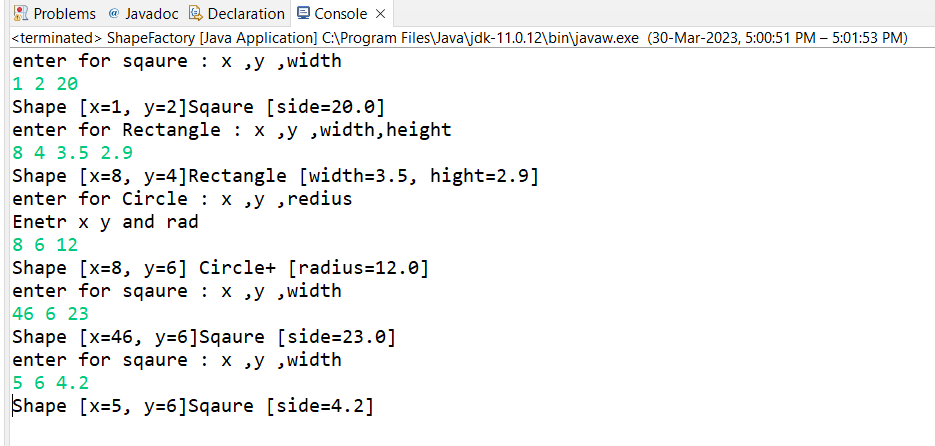
}

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P:-**



**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

**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**

7 Objective

: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);

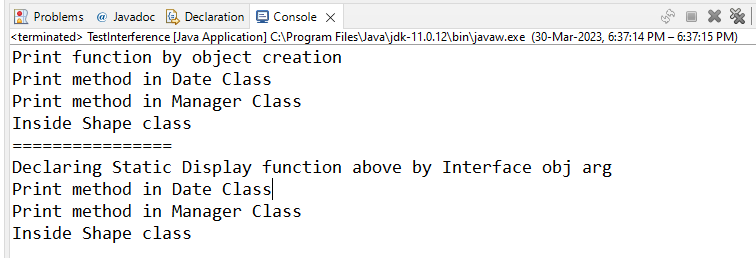
display(s);

}

}

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**O/P:-**



**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**