**Assignment Day 1**

1:write program to test Hello World.

**public** **class** Que1 {

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

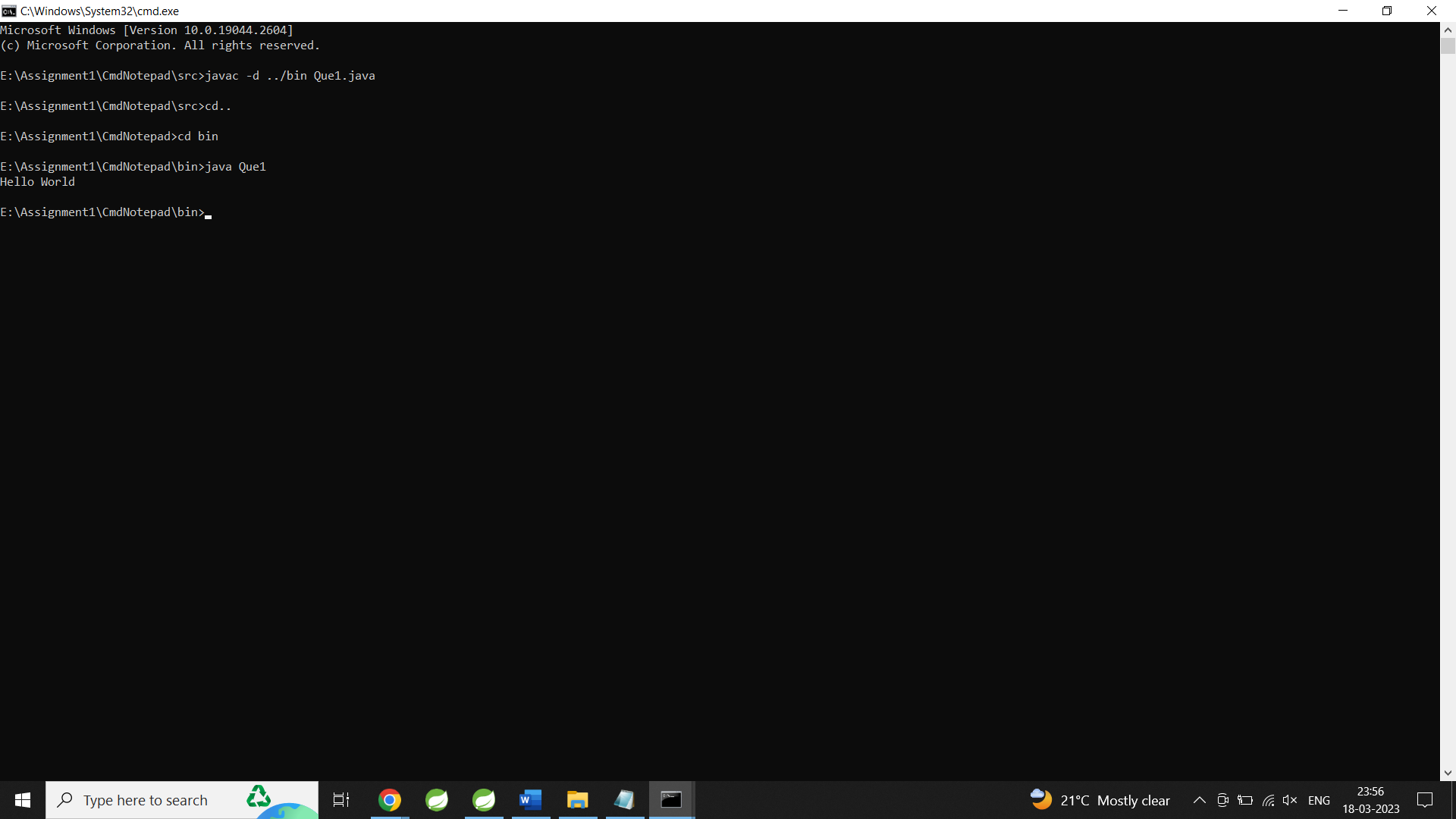
// **TODO** Auto-generated method stub

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

}

}

Output:



2:Write a program to adddition of two numbers

**import** java.util.Scanner;

**public** **class** Que2 {

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

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

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

System.***out***.println("Enter the First number: ");

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

System.***out***.println("Enter the Second number: ");

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

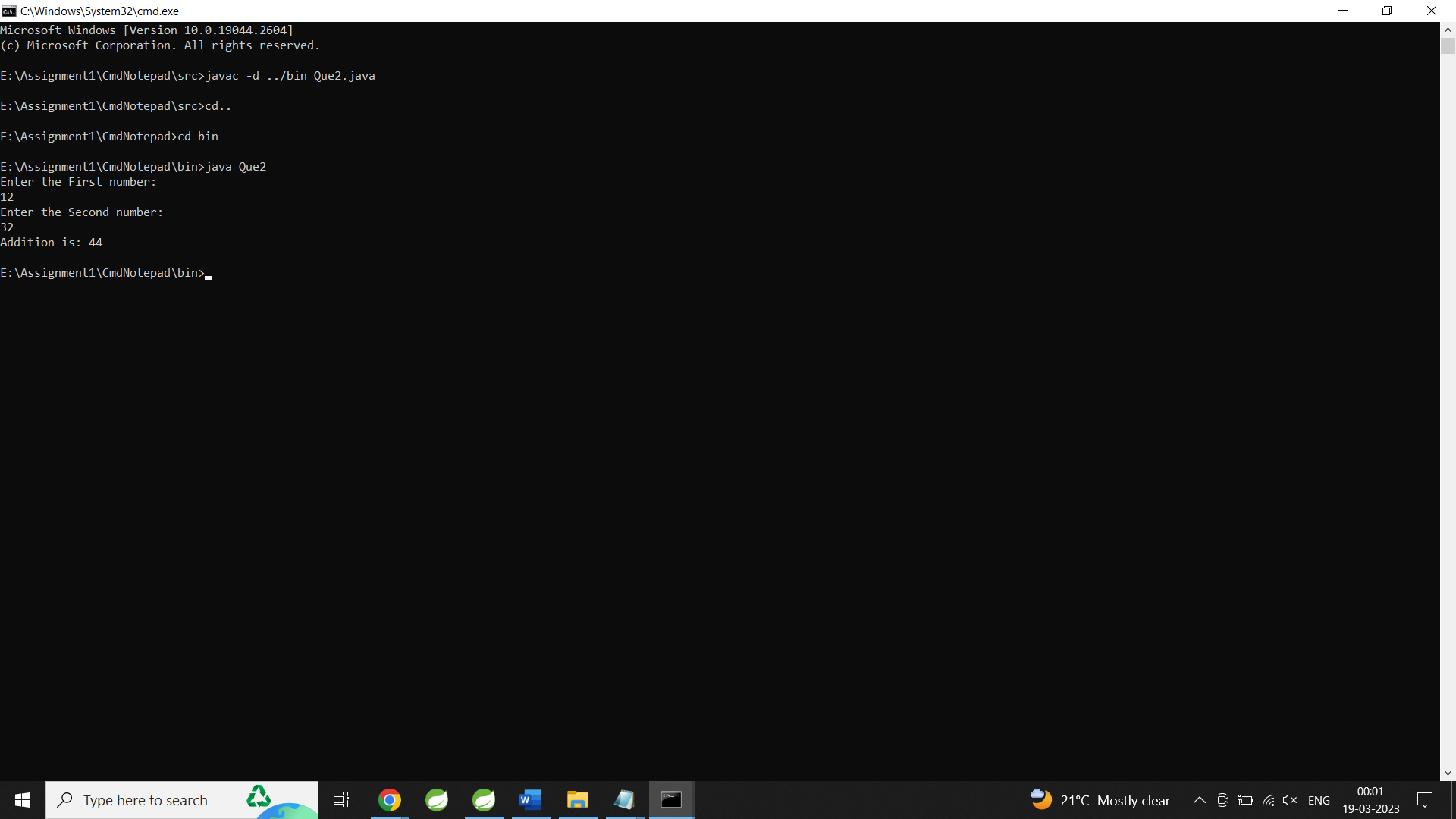
**int** c=a+b;

System.***out***.println("Addition is: "+c);

}

}

Output:



3:Write a program to swap two numbers.

**import** java.util.Scanner;

**public** **class** Que3 {

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

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

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

System.***out***.println("Enter the First number: ");

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

System.***out***.println("Enter the Second number: ");

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

System.***out***.println("Before Swap: ");

System.***out***.println("Num1=: "+a+" Num2: "+b);

**int** temp;

temp=a;

a=b;

b=temp;

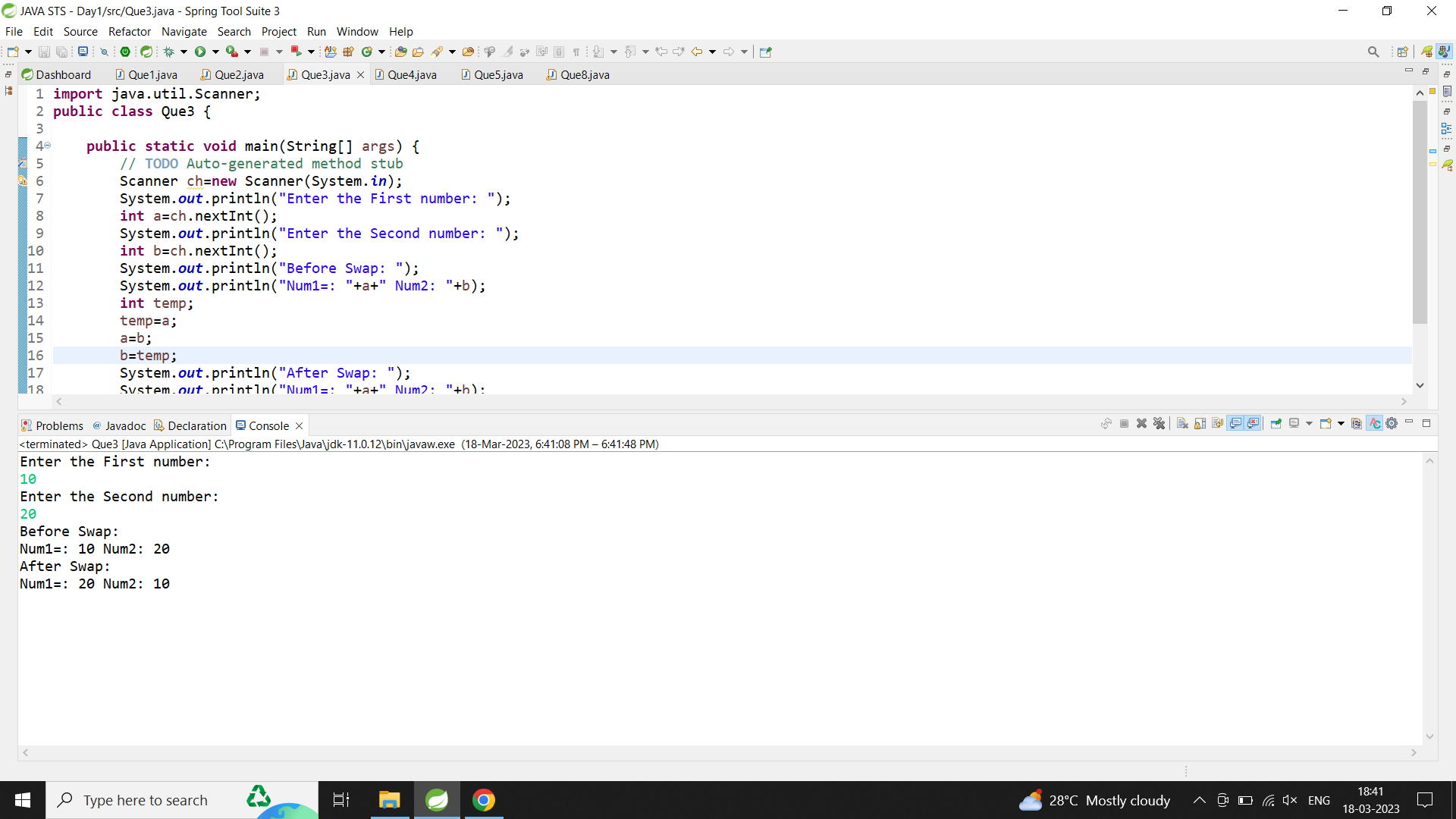
System.***out***.println("After Swap: ");

System.***out***.println("Num1=: "+a+" Num2: "+b);

}

}

Output:



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

**import** java.util.Scanner;

**public** **class** Que4 {

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

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

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

System.***out***.println("Enter the number: ");

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

**int** fact=1;

**for**(**int** i=1;i<=a;i++)

{

fact=fact\*i;

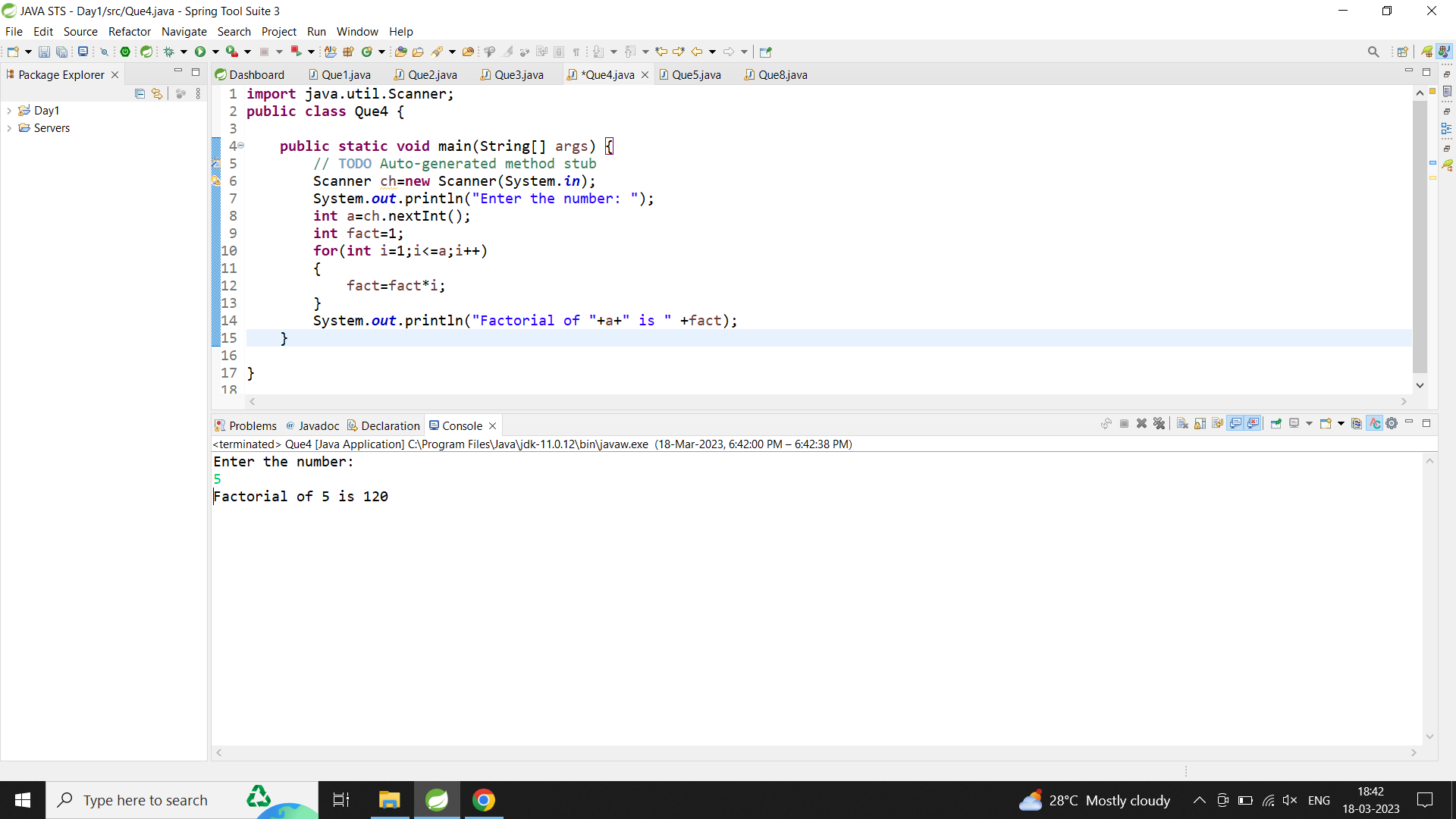
}

System.***out***.println("Factorial of "+a+" is " +fact);

}

}

Output:



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

**import** java.util.Scanner;

**public** **class** Que8 {

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

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

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

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

**int** n=ch.nextInt();

**int** even=0;

**int** odd=0;

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

{

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

even+=i;

**else**

odd+=i;

}

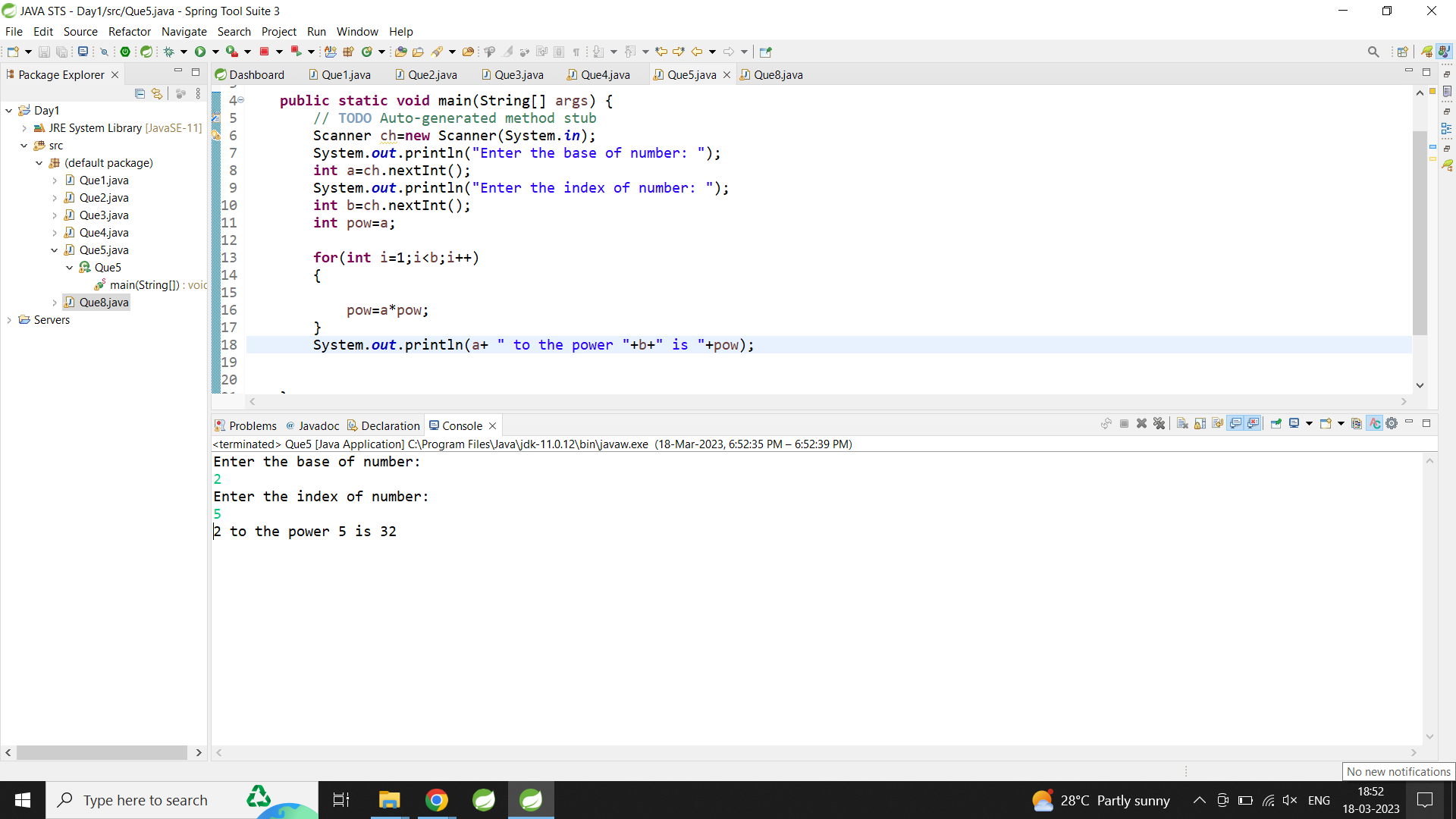
System.***out***.println("Sum of Even Numbers: "+even);

System.***out***.println("Sum of Odd numbers: "+odd);

}

}

Output:



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

**import** java.util.Scanner;

**public** **class** Que6 {

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

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

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

**int** n,flag=0;

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

n=sc.nextInt();

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

{

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

{

flag=1;

**break**;

}

}

**if**(flag==1)

{

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

}

**else**

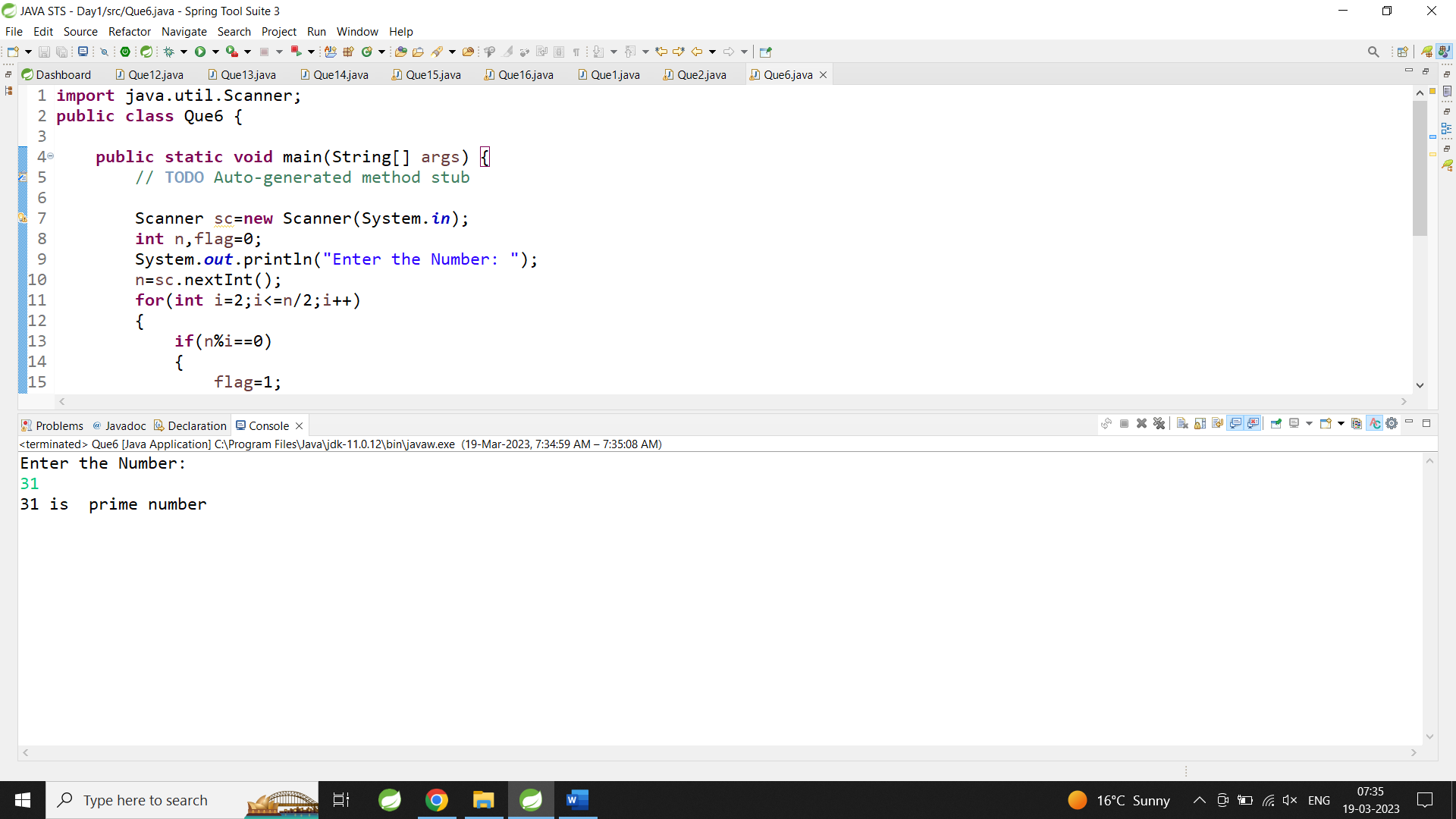
{

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

}

}

}



7:Sum of series :

1+2+3+….+n

**import** java.util.Scanner;

**public** **class** Que7 {

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

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

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

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

**int** n=ch.nextInt();

**int** sum=0;

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

{

sum=sum+i;

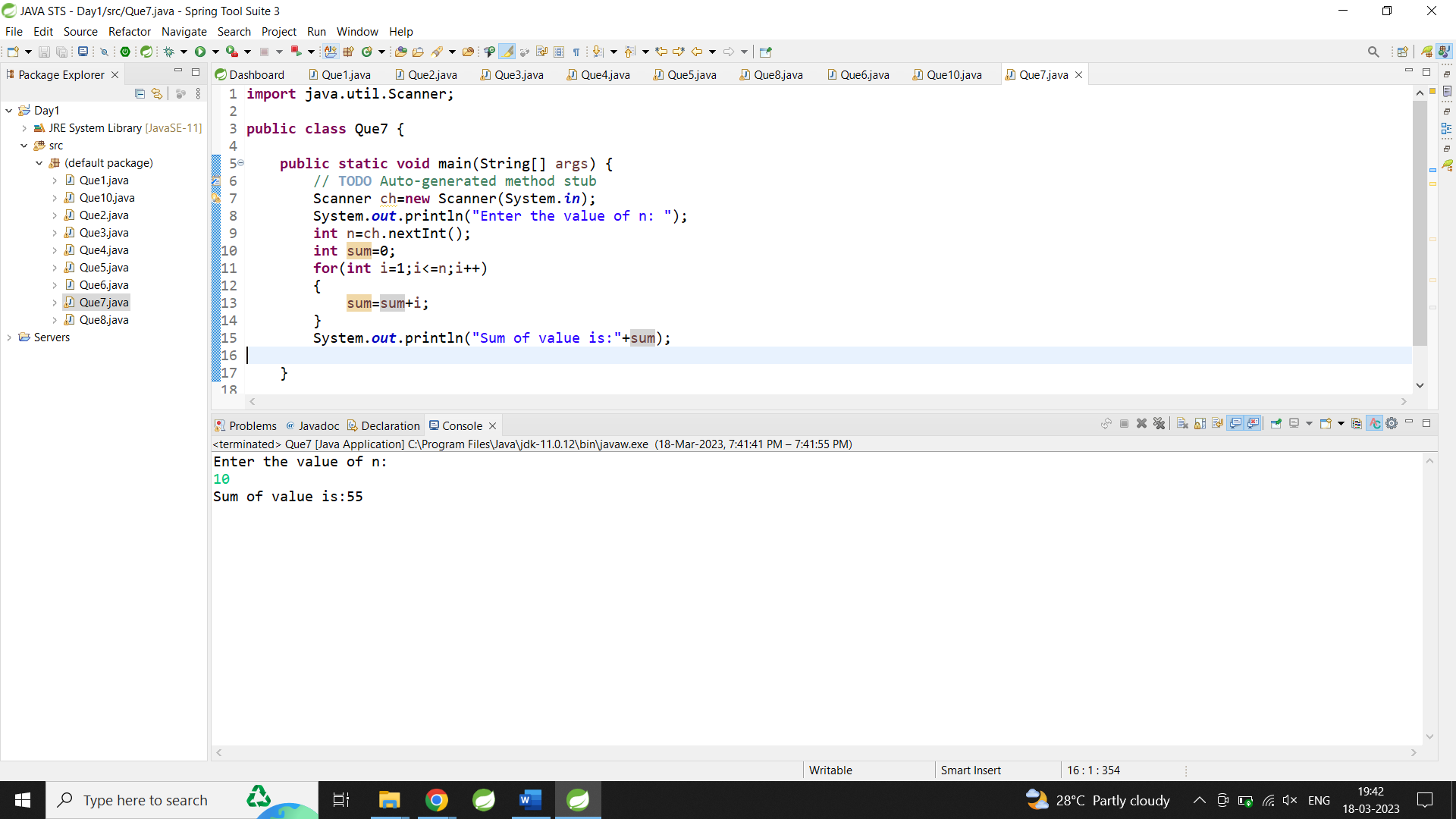
}

System.***out***.println("Sum of value is:"+sum);

}

}

Output:



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

**import** java.util.Scanner;

**public** **class** Que8 {

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

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

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

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

**int** n=ch.nextInt();

**int** even=0;

**int** odd=0;

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

{

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

even+=i;

**else**

odd+=i;

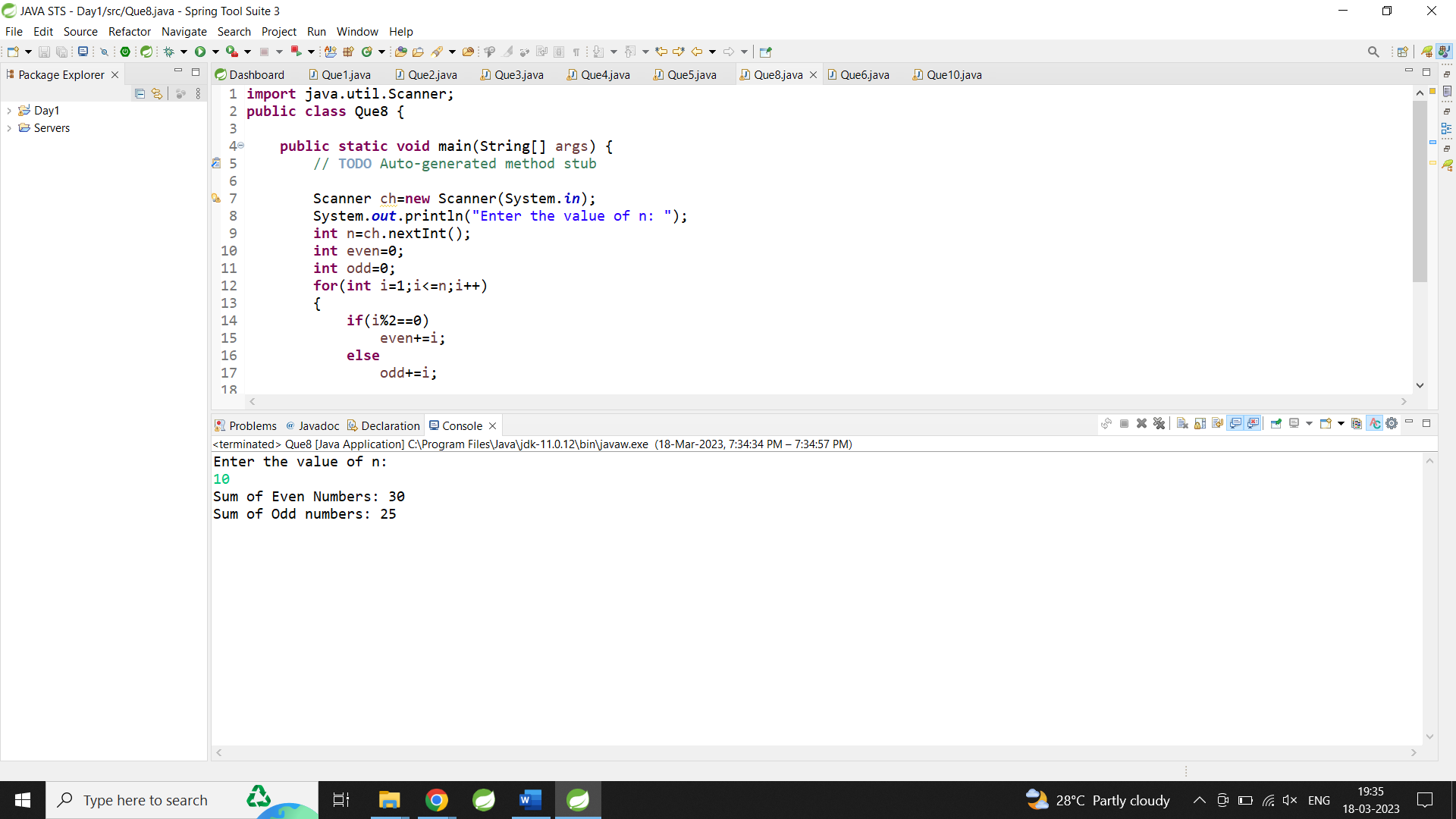
}

System.***out***.println("Sum of Even Numbers: "+even);

System.***out***.println("Sum of Odd numbers: "+odd);

}

}



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

**import** java.util.Scanner;

**public** **class** Que10 {

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

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

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

System.***out***.println("Enter the number: ");

**int** n=ch.nextInt();

**int** num,rev=0;

**while**(n>0)

{

num=n%10;

rev=rev\*10+num;

n=n/10;

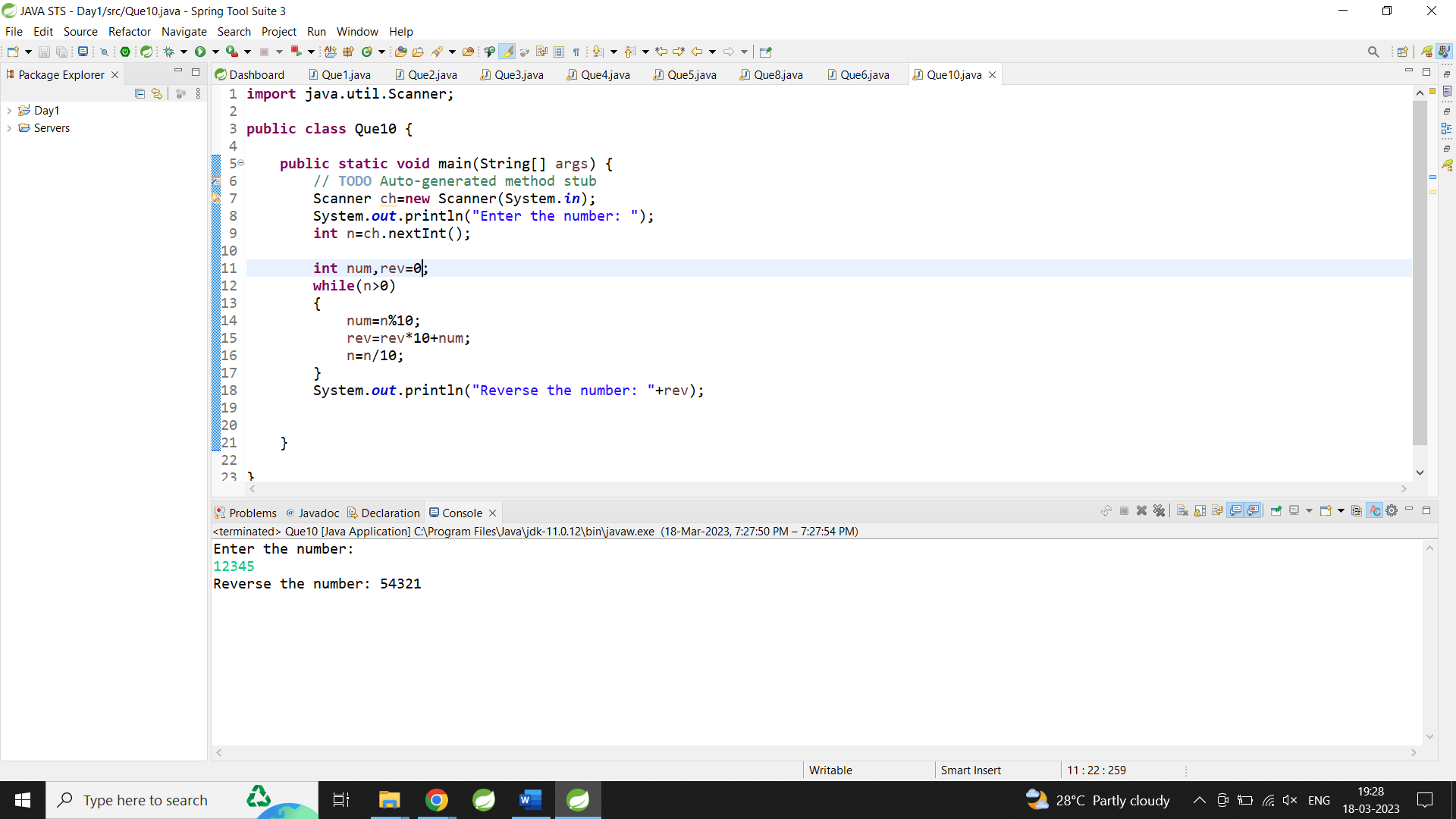
}

System.***out***.println("Reverse the number: "+rev);

}

}

Output:



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

**import** java.util.Scanner;

**public** **class** Prime {

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

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

System.***out***.println("Enter the range for prime munber:");

**int** flag = 0;

**int** j;

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

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

**for** (j = 2; j <= i / 2; j++) {

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

flag = 1;

**break**;

} **else** {

flag = 0;

}

}

**if** (flag == 0 && i != j) {

System.***out***.print(i + ", ");

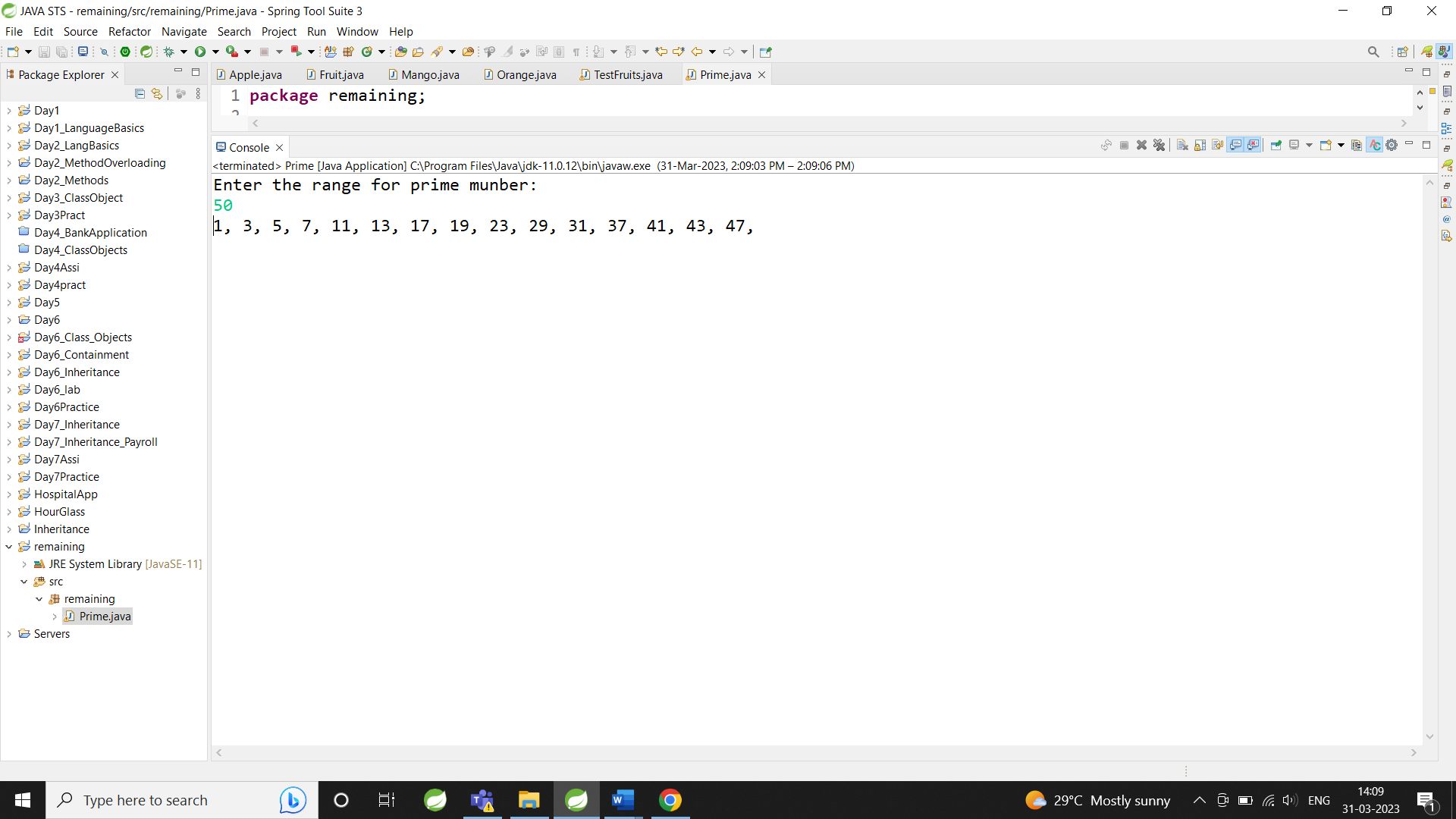
}

}

}

}

Output:



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

**import** java.util.Scanner;

**public** **class** Que12 {

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

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

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

System.***out***.println("Enter the number: ");

**int** n=ch.nextInt();

**int** key=n;

**int** num;

**int** sum=0;

**while**(n>0)

{

num=n%10;

sum=sum+(num\*num\*num);

n=n/10;

}

**if**(key==sum)

System.***out***.println(key+ " is armstrong Number");

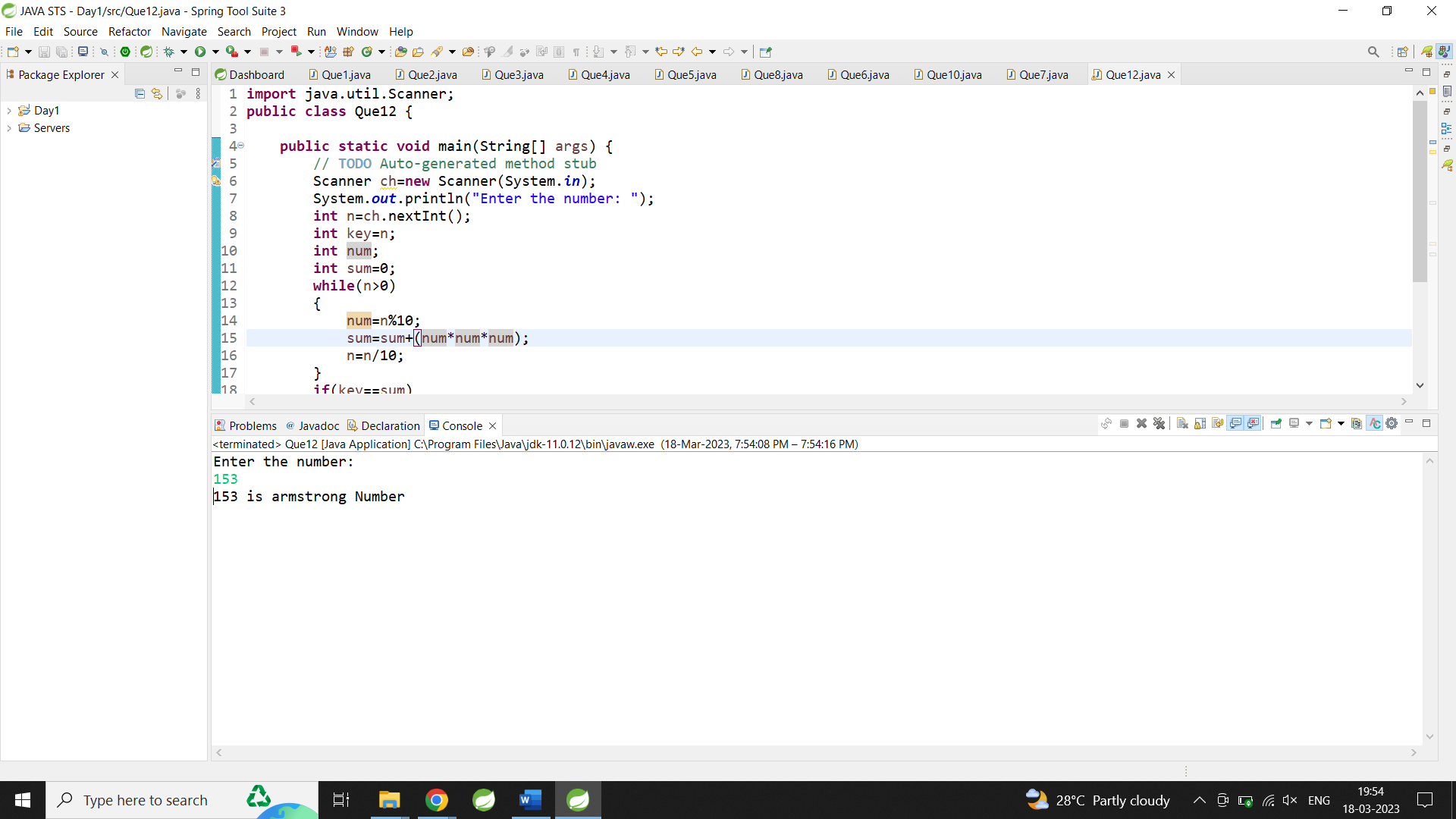
**else**

System.***out***.println(key+ " is not armstrong Number");

}

}

Output:



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

**import** java.util.Scanner;

**public** **class** Que13 {

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

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

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

System.***out***.println("Enter the First number: ");

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

System.***out***.println("Enter the Second number: ");

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

System.***out***.println("Enter the Second number: ");

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

**if**(a>b)

{

**if**(a>c) {

System.***out***.println(a+ " is greater");

}

**else**

{

System.***out***.println(c+ " is greater");

}

}

**else** {

**if**(b>c) {

System.***out***.println(b+ " is greater");

}

**else**

{

System.***out***.println(c+ " is greater");

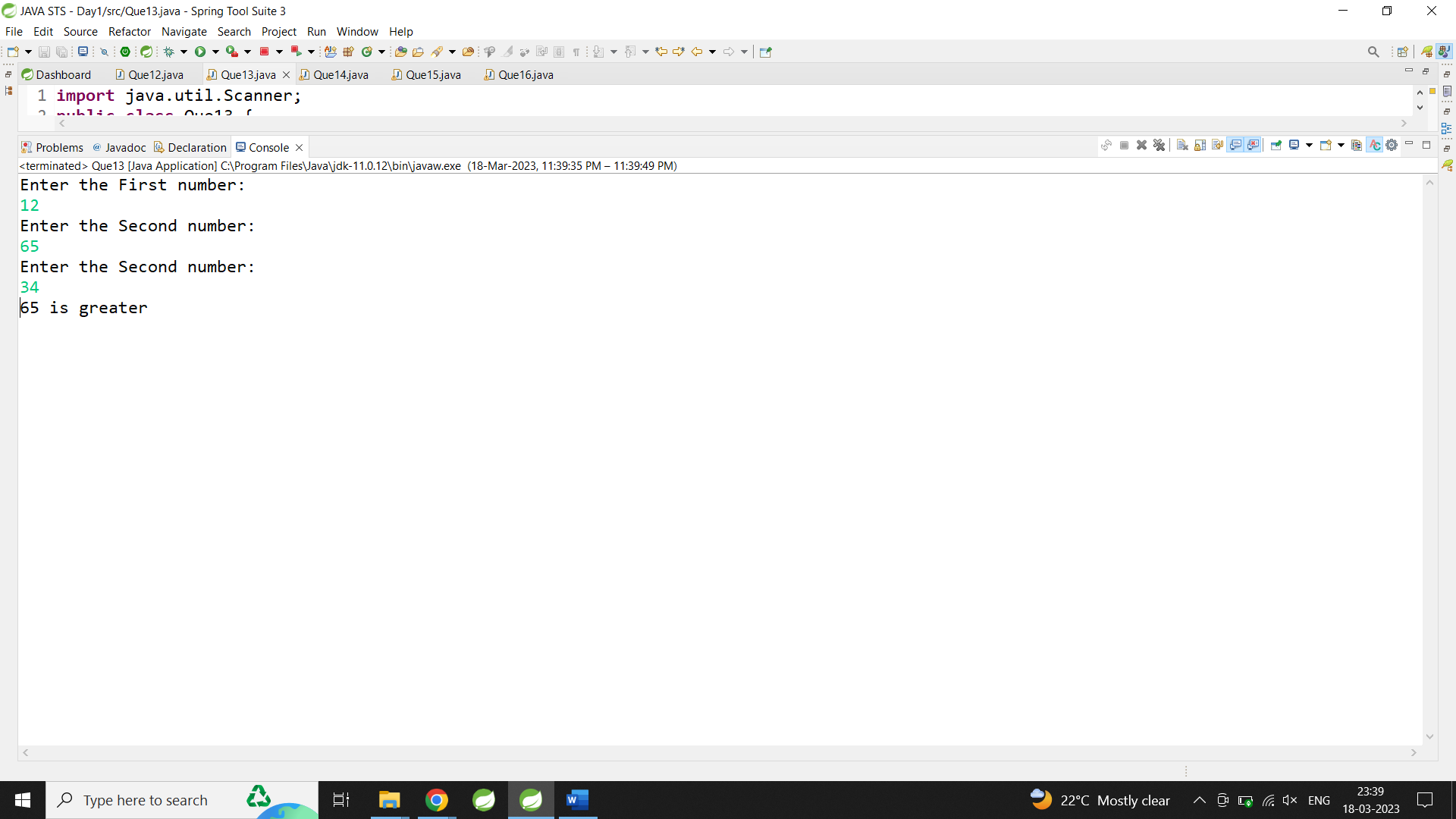
}

}

}

}

Output:



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

**import** java.util.Scanner;

**public** **class** Que14 {

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

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

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

**do** {

System.***out***.println("1.Pizza 200\n2.Cold Drink

50rs\n3.Garlic Bread 100rs\n4.Burger 70rs\n5.Total

Amount\n6.exit");

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

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

n=sc.nextInt();

**switch**(n)

{

**case** 1:System.***out***.println("Enter the number of Pizza: ");

i=sc.nextInt();

amount+=i\*200;

**break**;

**case** 2:

System.***out***.println("Enter the number of Cold drink: ");

i=sc.nextInt();

amount+=i\*50;

**break**;

**case** 3:

System.***out***.println("Enter the number of Garlic bread: ");

i=sc.nextInt();

amount+=i\*100;

**break**;

**case** 4:

System.***out***.println("Enter the number of Burger: ");

i=sc.nextInt();

amount+=i\*70;

**break**;

**case** 5:System.***out***.println("Total Amount is "+amount);

**break**;

**case** 6:System.***out***.println("Exit");

**break**;

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

**break**;

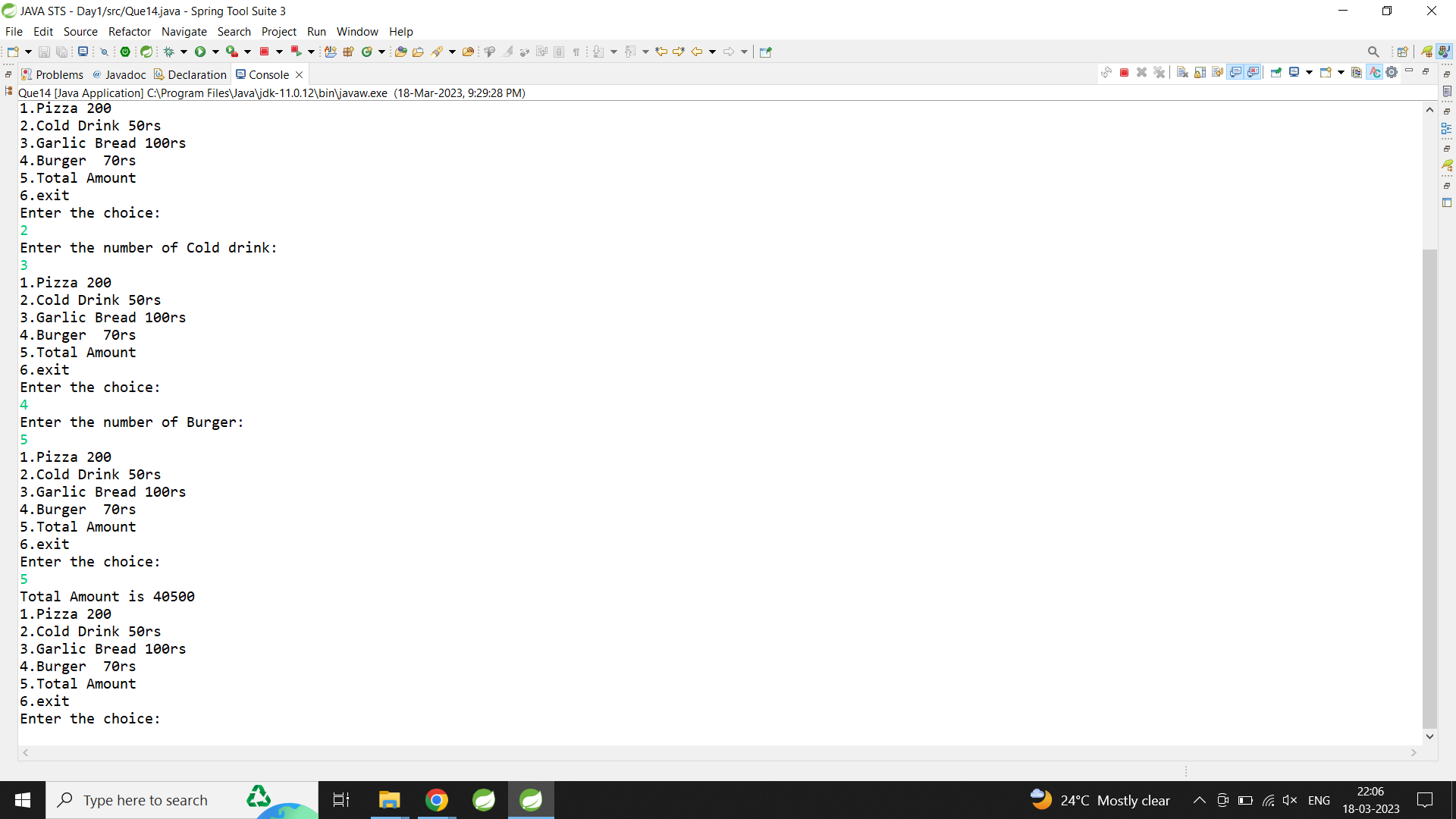
}

}**while**(n!=6);

}

}

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 array6:sum of array element

**import** java.util.Scanner;

**public** **class** Que15 {

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

**int** n, ch;

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

System.***out***.println("Enter value of n: ");

n = sc.nextInt();

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

**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 the choice: ");

ch = sc.nextInt();

**switch** (ch) {

**case** 1:

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

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

arr[i] = sc.nextInt();

}

**break**;

**case** 2:

System.***out***.println("Array Elements are: ");

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

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

}

**break**;

**case** 3:

System.***out***.println("Enter the element you want to search: ");

**int** key;

**int** flag = 0;

key = sc.nextInt();

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

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

flag = 1;

**break**;

} **else** {

flag = 0;

}

}

**if** (flag == 1)

System.***out***.println(key + " is present in array");

**else**

System.***out***.println(key + " is not present in array");

**break**;

**case** 4:

System.***out***.println("Reverse array: ");

**for** (**int** i = n - 1; i >= 0; i--) {

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

}

**break**;

**case** 5:

System.***out***.println("Even Elements are: ");

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

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

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

}

}

**break**;

**case** 6:

System.***out***.println("Sum of the array element is:");

**int** sum = 0;

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

sum = sum + arr[i];

}

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

**break**;

**case** 7:

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

**break**;

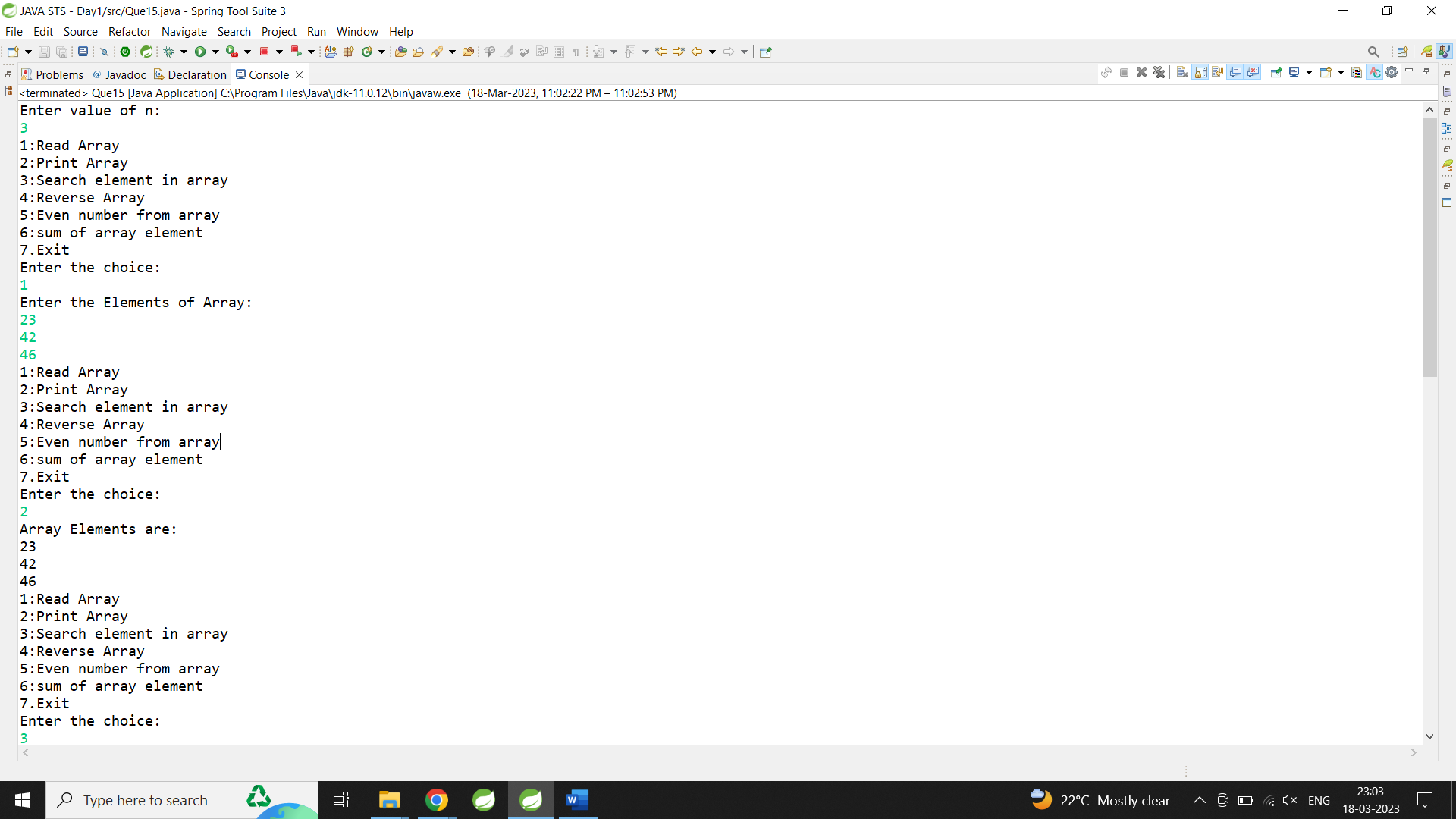
}

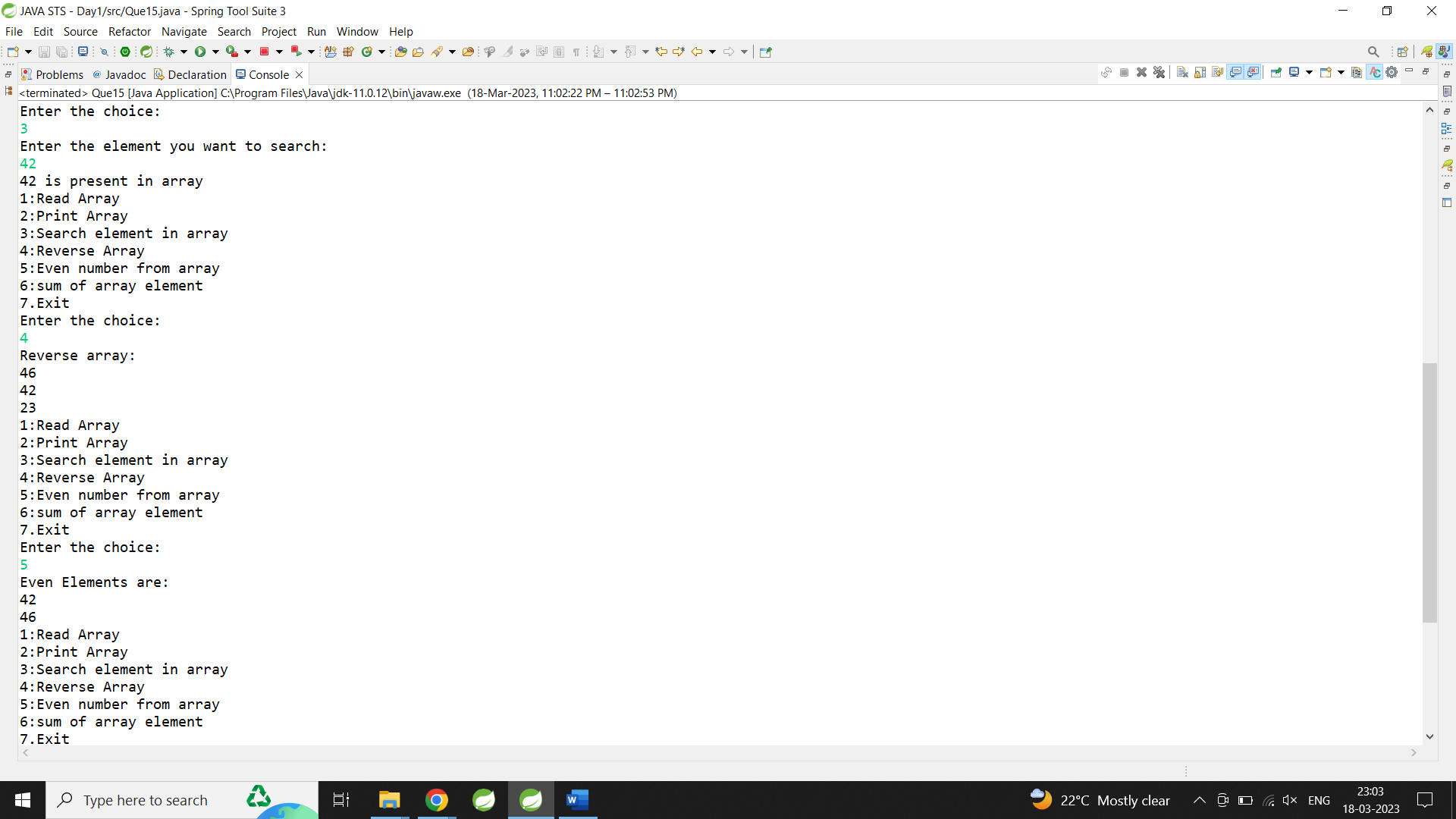
} **while** (ch != 7);

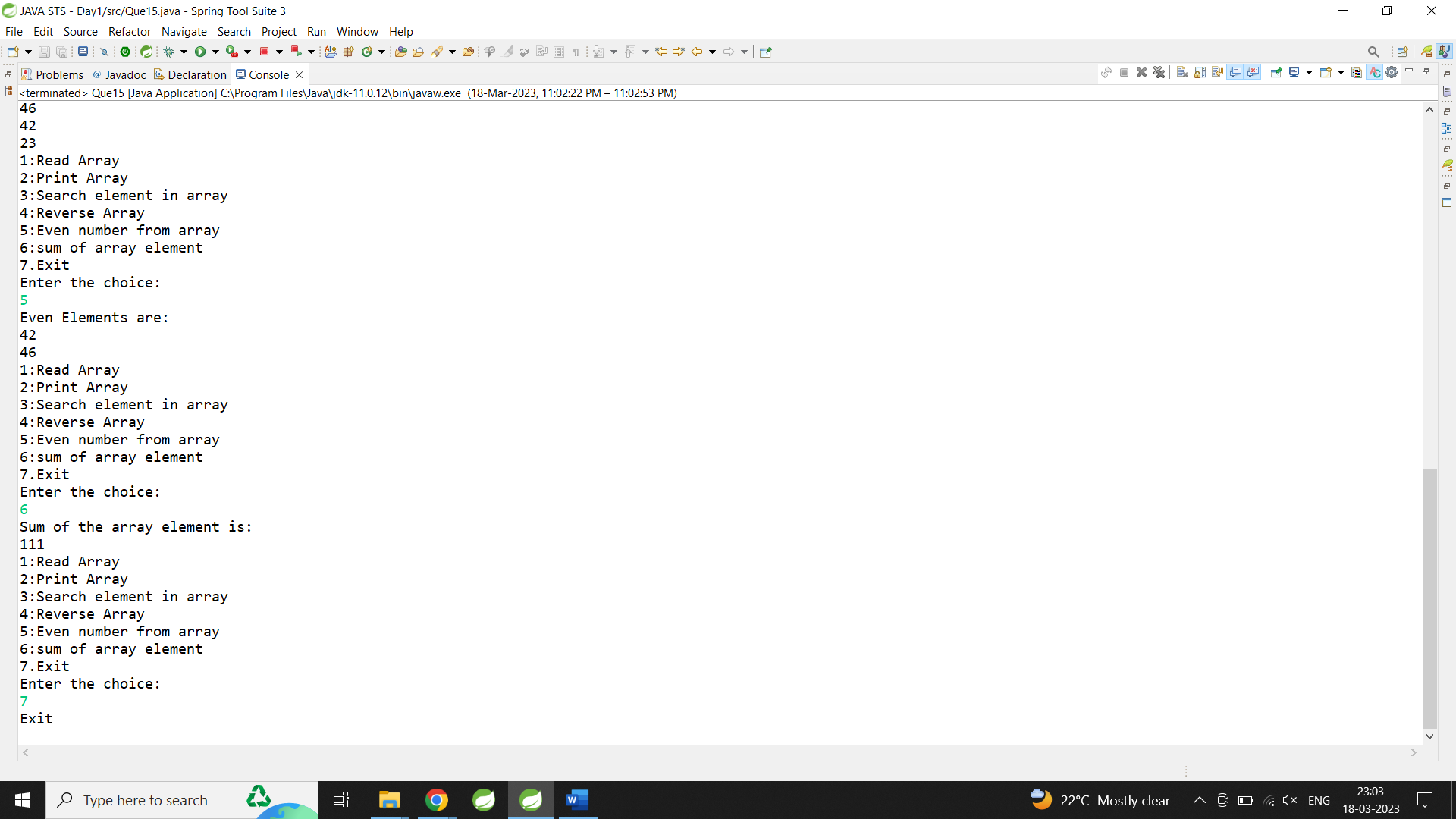
}

}

Output:







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

1 2 3

5 6 7 8 9

1 2 3 5 6 7 8 9

**import** java.util.Scanner;

**public** **class** Que16 {

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

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

System.***out***.println("Enter the value of n for Array1: ");

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

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

System.***out***.println("Enter the value of n for Array2: ");

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

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

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

**int** arr3[]=**new** **int**[10];

System.***out***.println("Enter the elements of for Array1: ");

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

{

arr1[i]=sc.nextInt();

}

System.***out***.println("Enter the elements of for Array2: ");

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

{

arr2[i]=sc.nextInt();

}

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

{

arr3[i]=arr1[i];

}

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

{

arr3[a+i]=arr2[i];

}

System.***out***.println("Elements of array1 and array2 are");

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

{

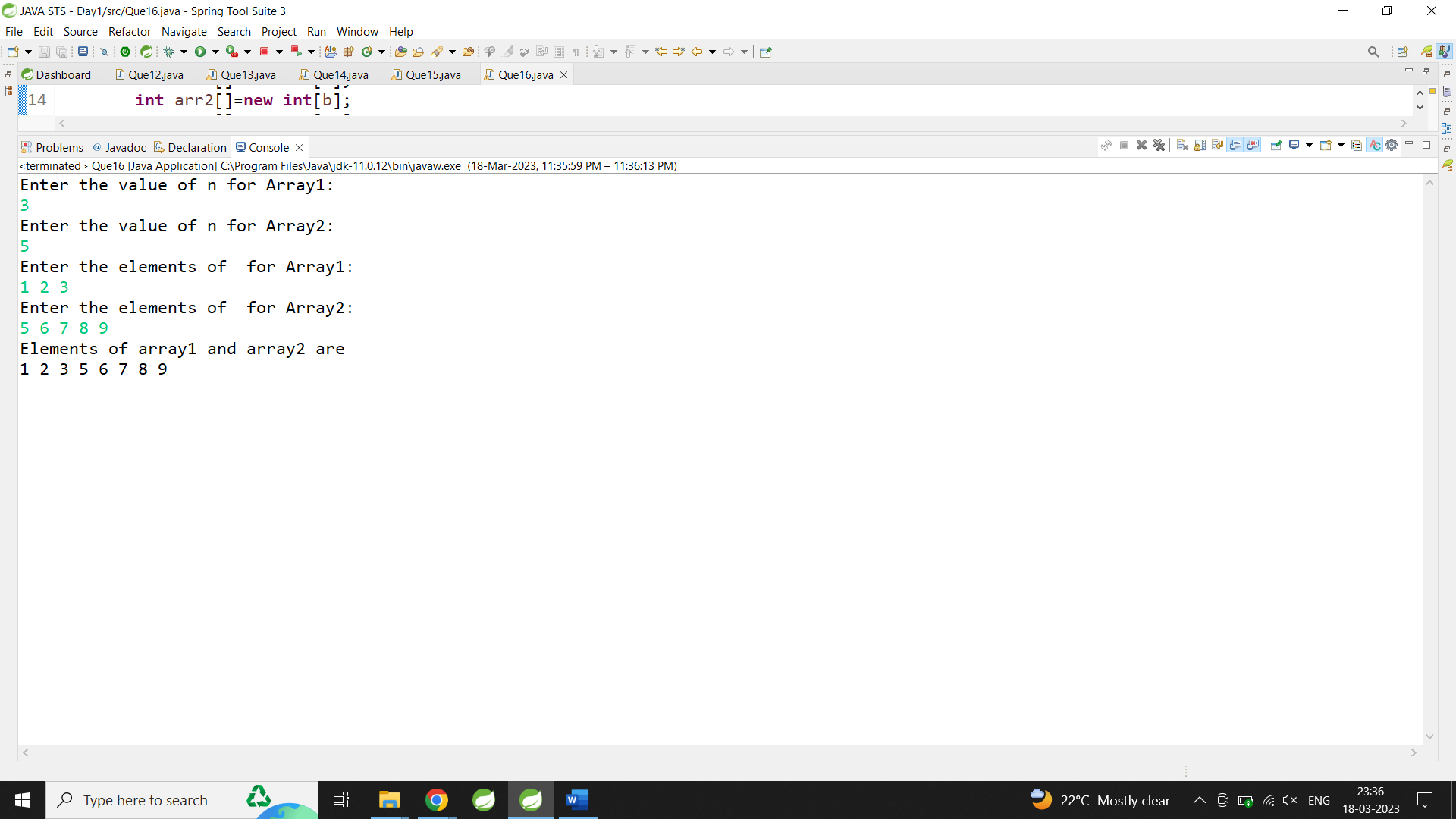
System.***out***.print(arr3[i]+ " ");

}

}

}

Output:



**Assignment Day 3**

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.

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

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

**public** Date()

{

day=10;

month=11;

year=2001;

}

**public** Date(**int** d,**int** mth,**int** yr)

{

day=d;

month=mth;

year=yr;

}

**public** **void** Display()

{

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

}

}

**public** **class** Que1 {

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

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

Date d=**new** Date();

d.Display();

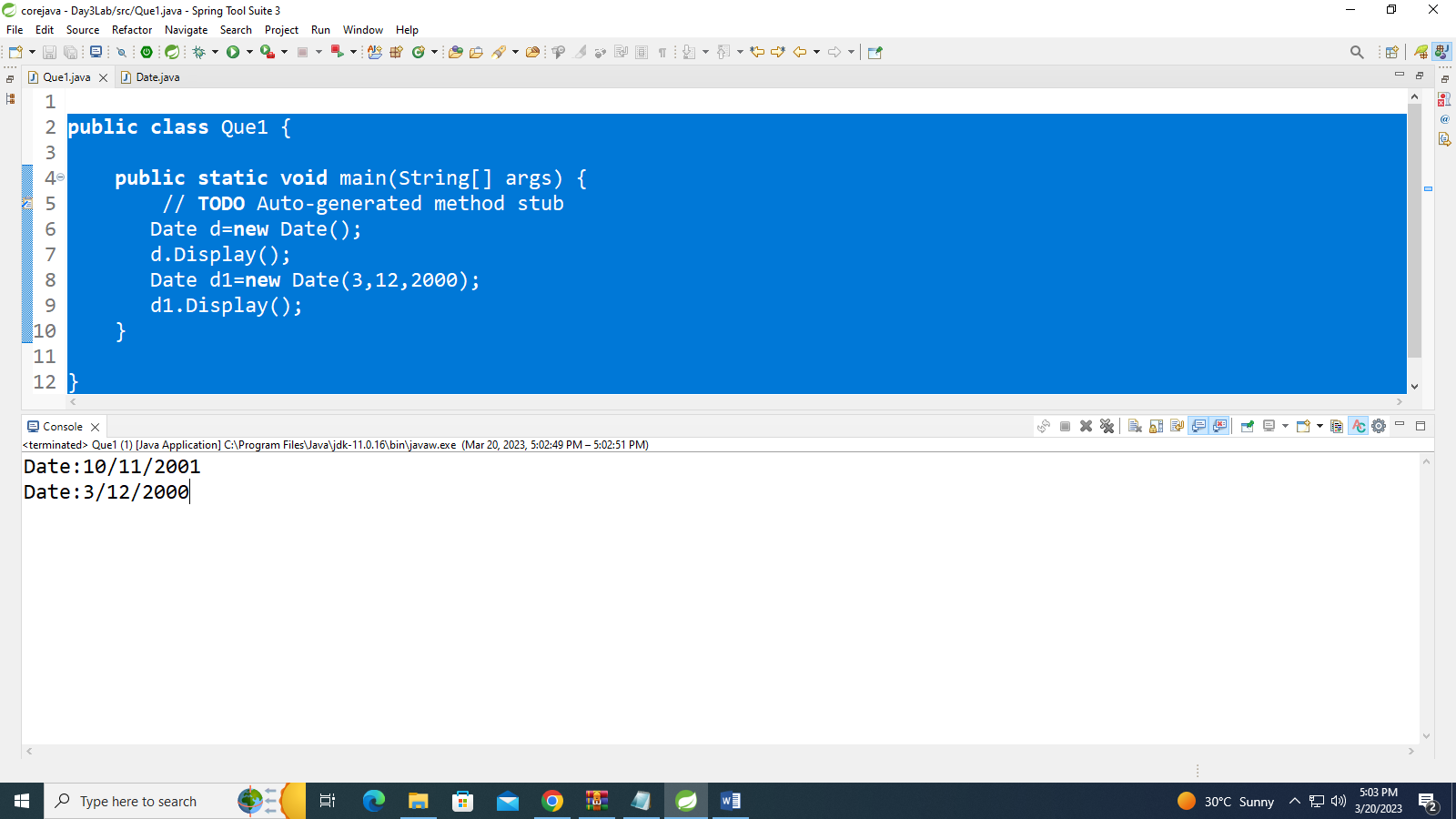
Date d1=**new** Date(3,12,2000);

d1.Display();

}

}

Output:



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.

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

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

**public** Date()

{

day=10;

month=11;

year=2001;

}

**public** Date(**int** d,**int** mth,**int** yr)

{

day=d;

month=mth;

year=yr;

}

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

{

day=d;

month=m;

year=y;

}

**public** **void** show()

{

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

}

}

**public** **class** Employee {

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

**private** String name;

**private** Date dob;

**public** Employee()

{

empid=101;

name="Piyush";

dob=**new** Date();

}

**public** Employee(**int** i,String n,Date d)

{

empid=i;

name=n;

dob=d;

}

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

{

System.***out***.println("Empid="+empid+"\nName="+name);

dob.show();

}

}

**import** java.util.Scanner;

**public** **class** Que2 {

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

//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.

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

System.***out***.println("Enter the date: Day/Month/Year:");

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

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

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

Date d1=**new** Date(12,11,2000);

Employee e1=**new** Employee(12,"Anshuja",d1);

e1.display();

Date d2=**new** Date(d,m,y);

Employee e2=**new** Employee(23,"pratiksha", d2);

e2.display();

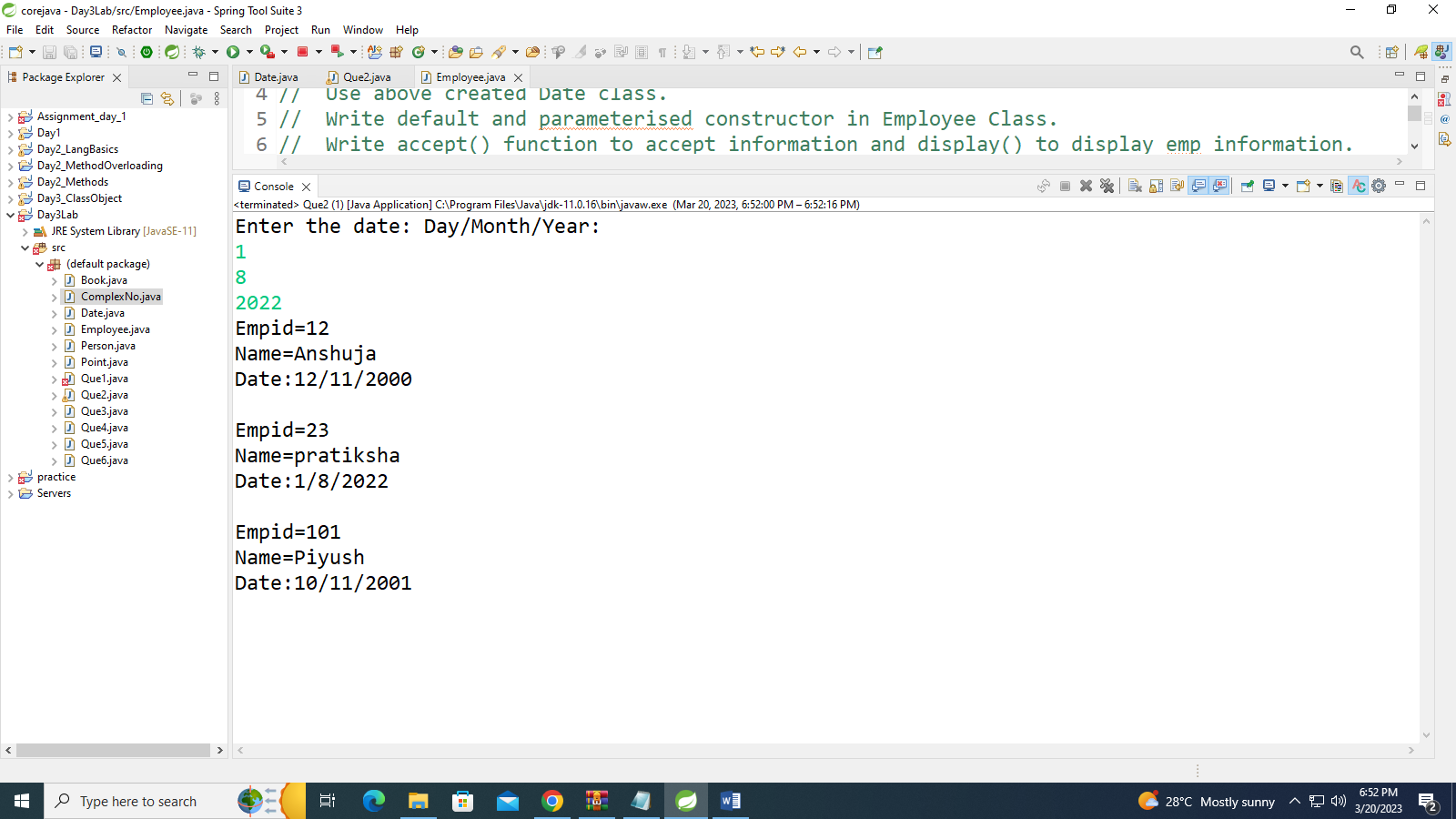
Employee e3=**new** Employee();

e3.display();

}

}

Output:



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.

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

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

**private** String name,city;

**public** Person()

{

age=21;

name="Pratiksha";

city="Dhule";

}

**public** Person(**int** ag,String nm, String ct)

{

age=ag;

name=nm;

city=ct;

}

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

{

**this**.age=age;

}

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

{

**this**.name=name;

}

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

{

**this**.city=city;

}

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

{

**return** age;

}

**public** String getName()

{

**return** name;

}

**public** String getCity()

{

**return** city;

}

**public** **void** Display()

{

System.***out***.println("Age: "+age+"\nName:"+name+"\nCity: "+city+"\n");

}

}

**public** **class** Que3 {

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

Person p=**new** Person();

p.Display();

Person p1=**new** Person(24,"Sayali","Pune");

p1.Display();

System.***out***.println("After updating age name city:\n");

p1.setName("anshu");

p1.setAge(20);

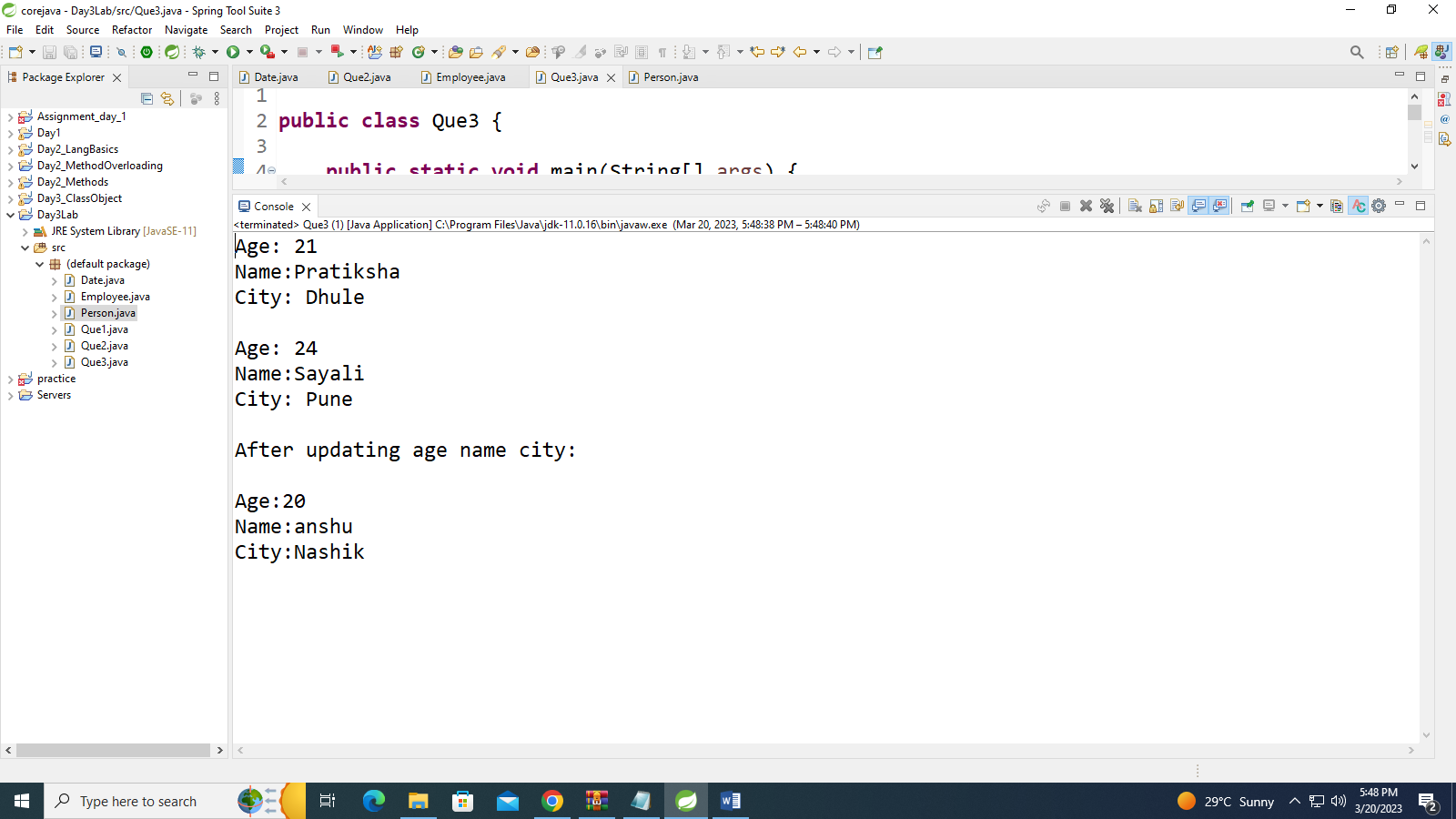
p1.setCity("Nashik");

System.***out***.println("Age:"+p1.getAge()+"\nName:"+p1.getName()+"\nCity:"+p1.getCity());

}

}

Output:



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.

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

**private** String name,author;

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

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

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

{

**this**.name=name;

}

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

{

**this**.author=author;

}

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

{

**this**.id=id;

}

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

{

**this**.price=price;

}

**public** String getName()

{

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

}

**public** String getAuthor()

{

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

}

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

{

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

}

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

{

**return** **this**.price;

}

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

{

System.***out***.println("Book id="+id+"\nName="+name+"\nAuthor="+author+"\nPrice="+price+"\n");

}

}

**public** **class** Que4 {

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

System.***out***.println("Before setter:");

Book b=**new** Book();

b.display();

System.***out***.println("After setters:");

b.setId(1);

b.setName("Java");

b.setAuthor("johny");

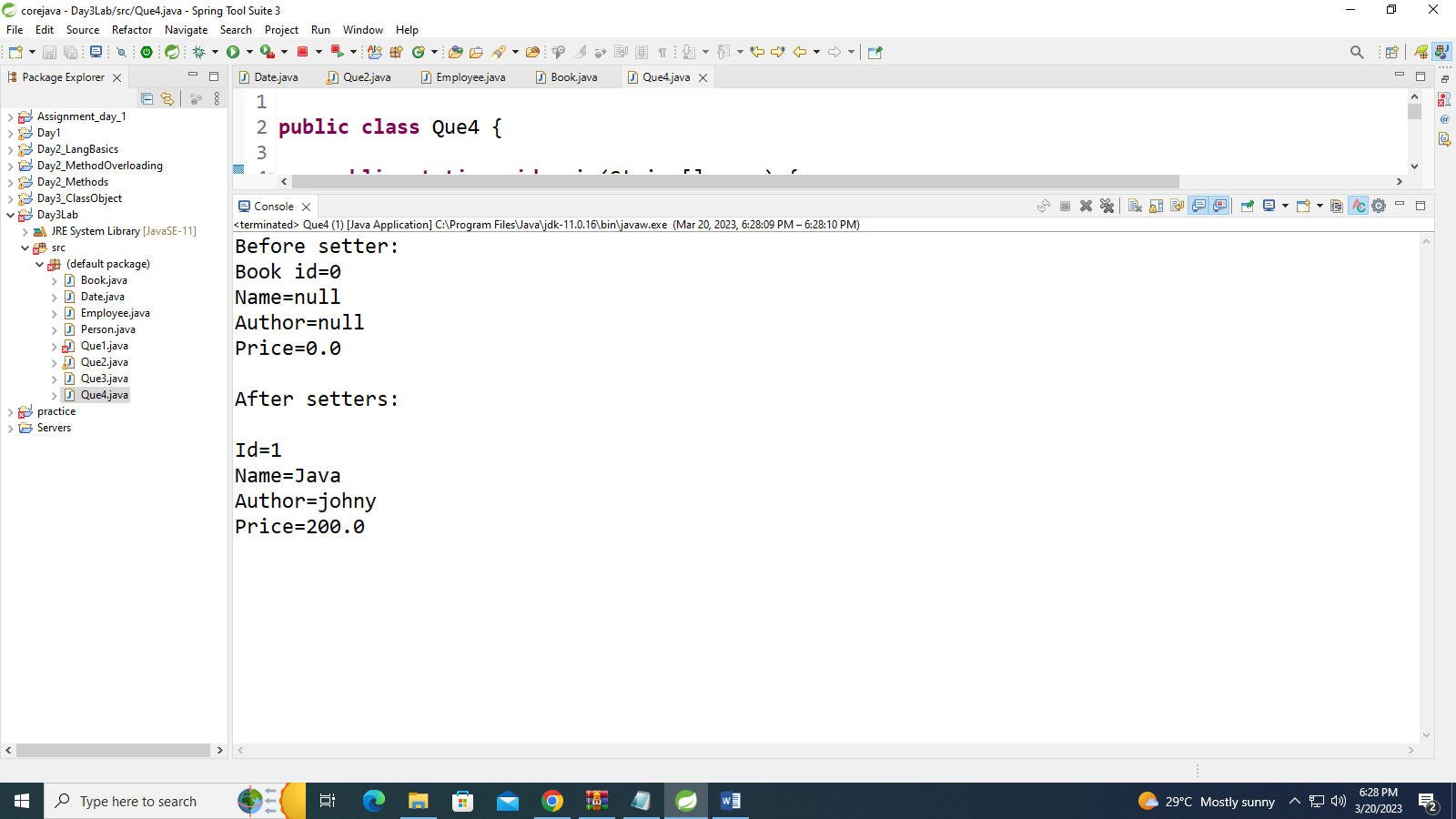
b.setPrice(200.00);

System.***out***.println("\nId="+b.getId()+"\nName="+b.getName()+"\nAuthor="+b.getAuthor()+"\nPrice="+b.getPrice());

}

}

Output:



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.

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

**private** **int** x,y;

**public** **void** setx(**int** x)

{

**this**.x=x;

}

**public** **void** sety(**int** y)

{

**this**.y=y;

}

**public** **int** getx()

{

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

}

**public** **int** gety()

{

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

}

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

{

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

}

}

**public** **class** Que5 {

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

System.***out***.println("Before setter:");

Point b=**new** Point();

b.display();

System.***out***.println("After setters:");

b.setx(10);

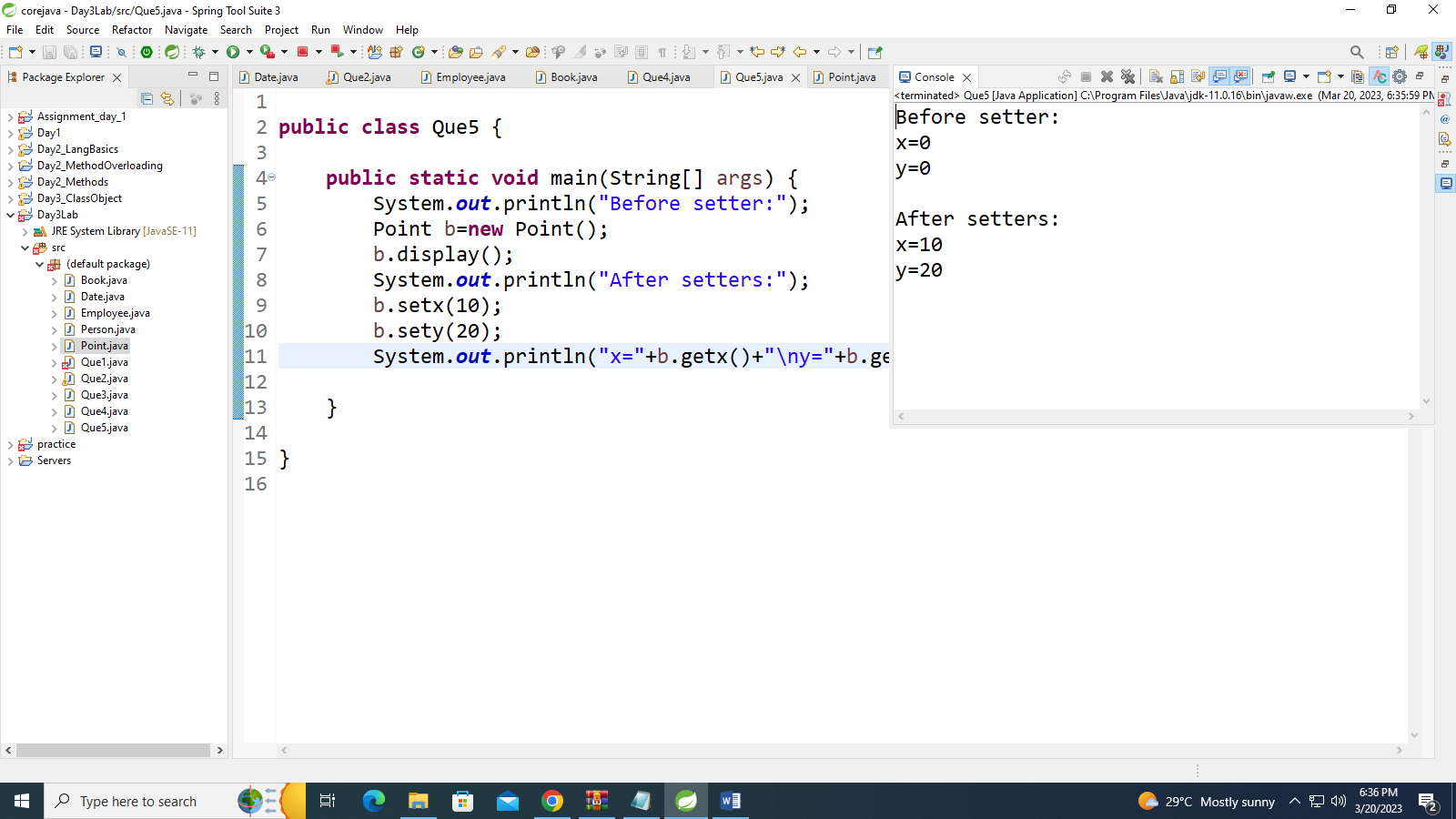
b.sety(20);

System.***out***.println("x="+b.getx()+"\ny="+b.gety());

}

}

Output:



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.

**public** **class** ComplexNo {

**private** **int** r,i;

**public** **void** setr(**int** r )

{

**this**.r=r;

}

**public** **void** seti(**int** i)

{

**this**.i=i;

}

**public** **int** getr()

{

**return** **this**.r;

}

**public** **int** geti()

{

**return** **this**.i;

}

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

{

System.***out***.println("complex no="+r+"+"+i+"i\n");

}

}

**public** **class** Que6 {

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

System.***out***.println("Before setter:");

ComplexNo b=**new** ComplexNo();

b.display();

System.***out***.println("After setters:");

b.setr(10);

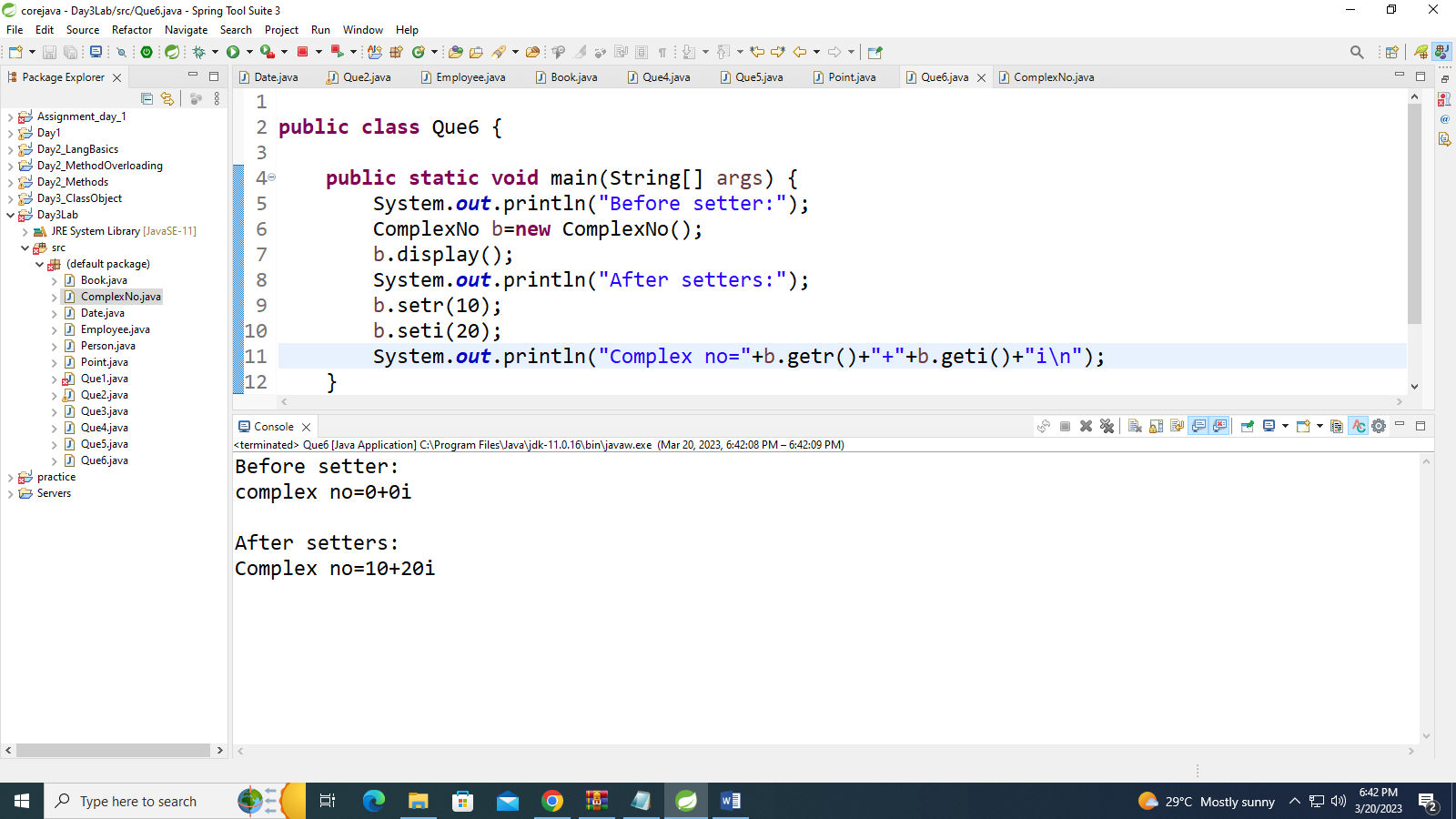
b.seti(20);

System.***out***.println("Complex no="+b.getr()+"+"+b.geti()+"i\n");

}

}

Output:



**Assignment Day 4**

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.

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

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

**private** **double** percentage;

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

**public** **void** acceptInfo(**int** rollno,**int** marks1,**int** marks2,**int** marks3)

{

**this**.rollno=rollno;

**this**.marks1=marks1;

**this**.marks2=marks2;

**this**.marks3=marks3;

}

**public** **void** calculate()

{

total=marks1+marks2+marks3;

percentage=total/3;

**if**(percentage>=80)

{

grade='A';

//System.out.println(grade);

}

**else** **if**(percentage<80 && percentage>75)

{

grade='B';

}

**else**

{

grade='c';

}

}

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

{

System.***out***.println("Roll no="+rollno+" marks1="+marks1+" marks2="+marks2+" marks3="+marks3);

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

System.***out***.println("Total="+total+"\nPercentage="+percentage+"\nGrade="+grade);

}

}

**import** java.util.Scanner;

**public** **class** StudentQue1 {

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

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

Student s=**new** Student();

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

System.***out***.println("Enter rollno, marks1,mark2,mark3:");

s.acceptInfo(sc.nextInt(),sc.nextInt(),sc.nextInt(),sc.nextInt());

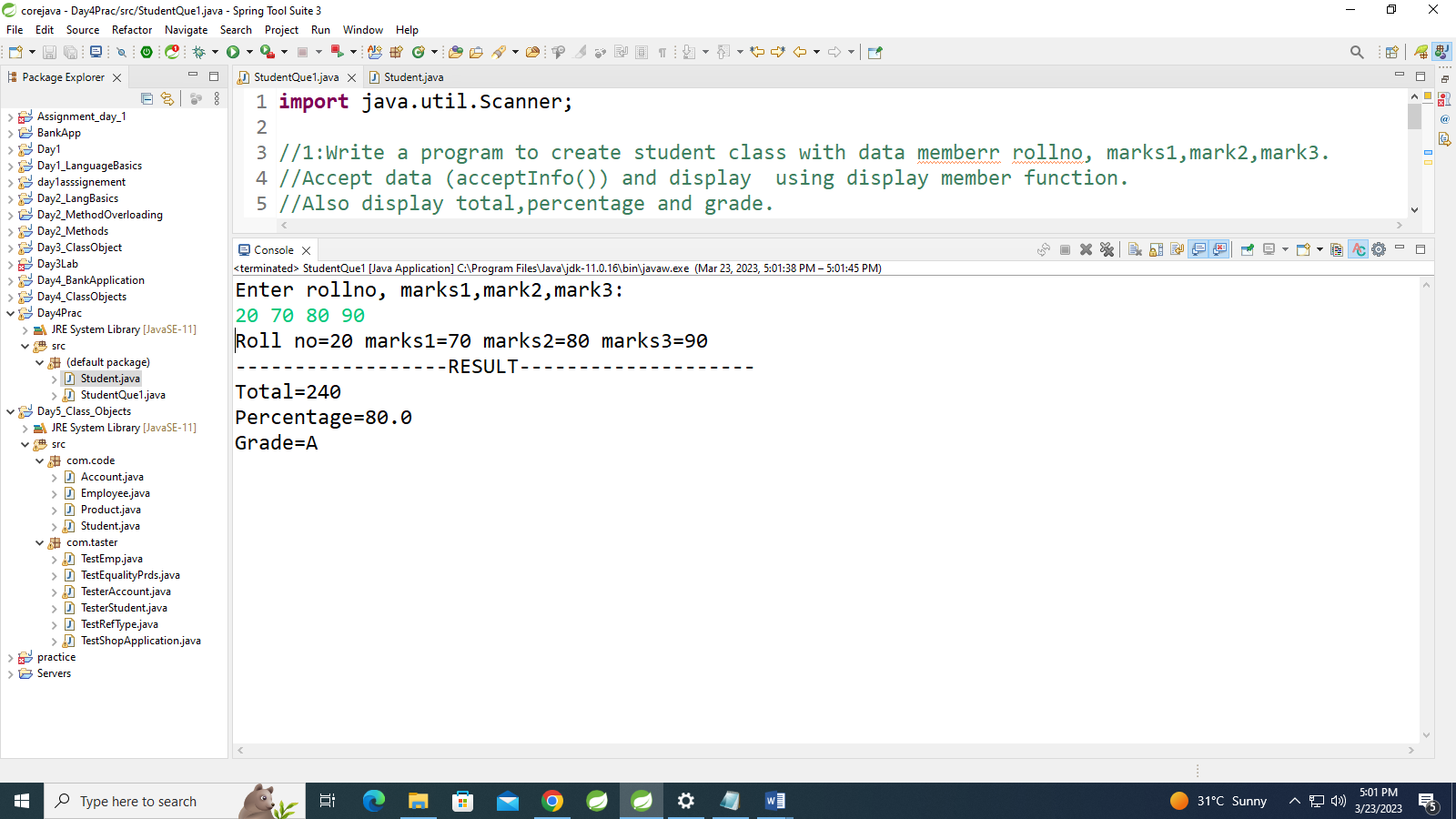
s.calculate();

s.display();

}

}

Output-



1. 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.

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

**private** String name,city;

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

**public** String getName() {

**return** name;

}

**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** Person() {

System.***out***.println("------------Inside Default constr-----------");

name = "Syali";

city = "pune";

age = 25;

}

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

System.***out***.println("------------Inside Parametrized constr-----------");

**this**.name = name;

**this**.city = city;

**this**.age = age;

}

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

{

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

}

}

**import** java.util.Scanner;

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

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

Person p=**new** Person();

p.display();

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

System.***out***.println("Enter name, city,age");

Person p1=**new** Person(sc.next(),sc.next(),sc.nextInt());

p1.display();

System.***out***.println("Enter the city want to change:");

p1.setCity(sc.next());

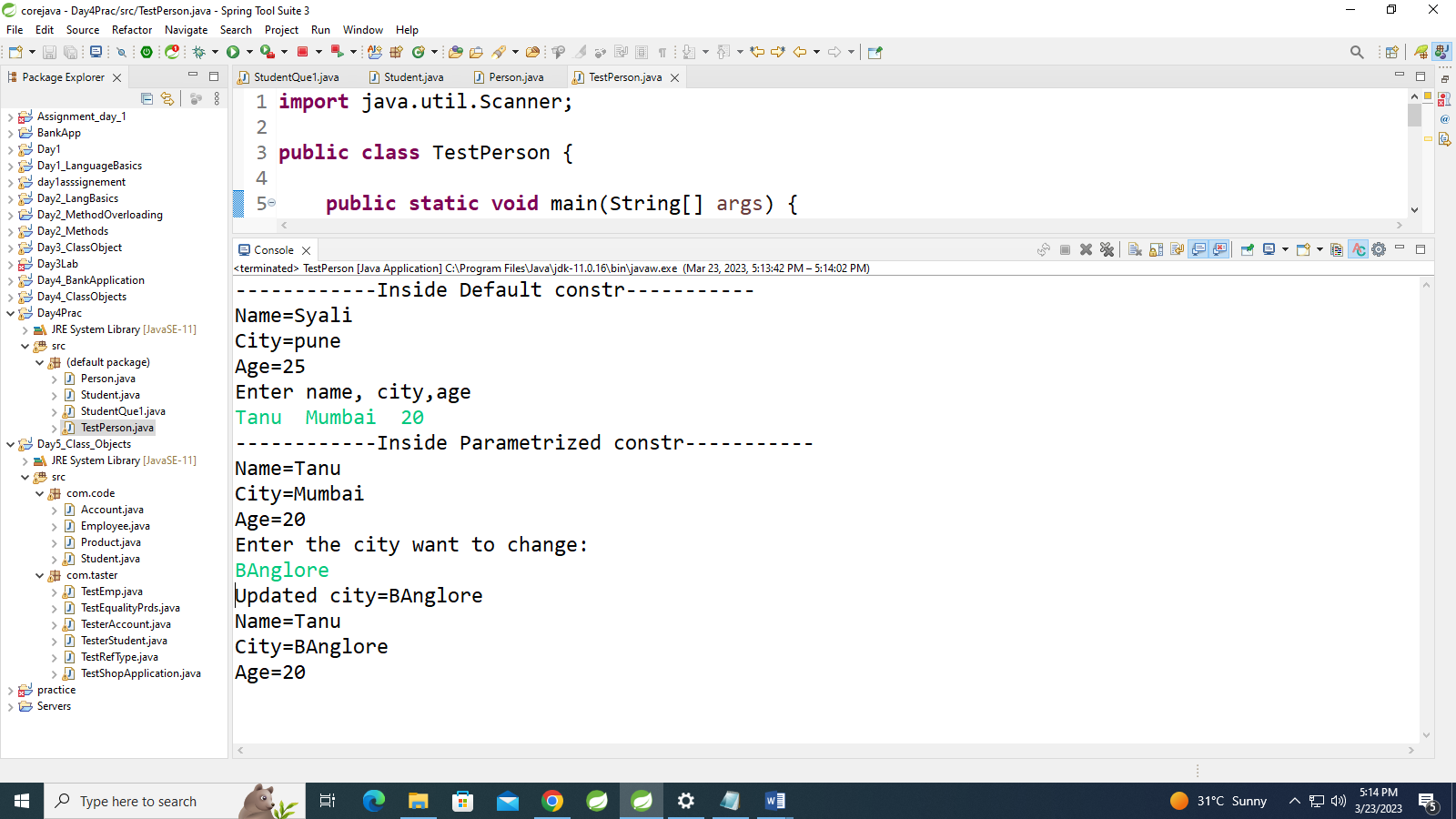
System.***out***.println("Updated city="+p1.getCity());

p1.display();

}

}

Output-



2. Create a class Date with data members as dd, mm, yy. 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.

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

**private** **int** dd, mm, yy;

**public** Date() {

System.***out***.println("------------Inside Default constr-----------");

dd = 10;

mm = 11;

yy = 2000;

}

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

System.***out***.println("------------Inside Parametrized constr-----------");

**this**.dd = dd;

**this**.mm = mm;

**this**.yy = yy;

}

**public** **int** getDd() {

**return** dd;

}

**public** **void** setDd(**int** dd) {

**this**.dd = dd;

}

**public** **int** getMm() {

**return** mm;

}

**public** **void** setMm(**int** mm) {

**this**.mm = mm;

}

**public** **int** getYy() {

**return** yy;

}

**public** **void** setYy(**int** yy) {

**this**.yy = yy;

}

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

System.***out***.println("Date:"+dd+"/"+mm+"/"+yy);

}

}

**import** java.util.Scanner;

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

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

Date d=**new** Date();

d.display();

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

System.***out***.println("Enter the date,month,year: ");

Date d2=**new** Date(s.nextInt(),s.nextInt(),s.nextInt());

d2.display();

System.***out***.println("Update the details:dd/mm/yy");

d2.setDd(s.nextInt());

d2.setMm(s.nextInt());

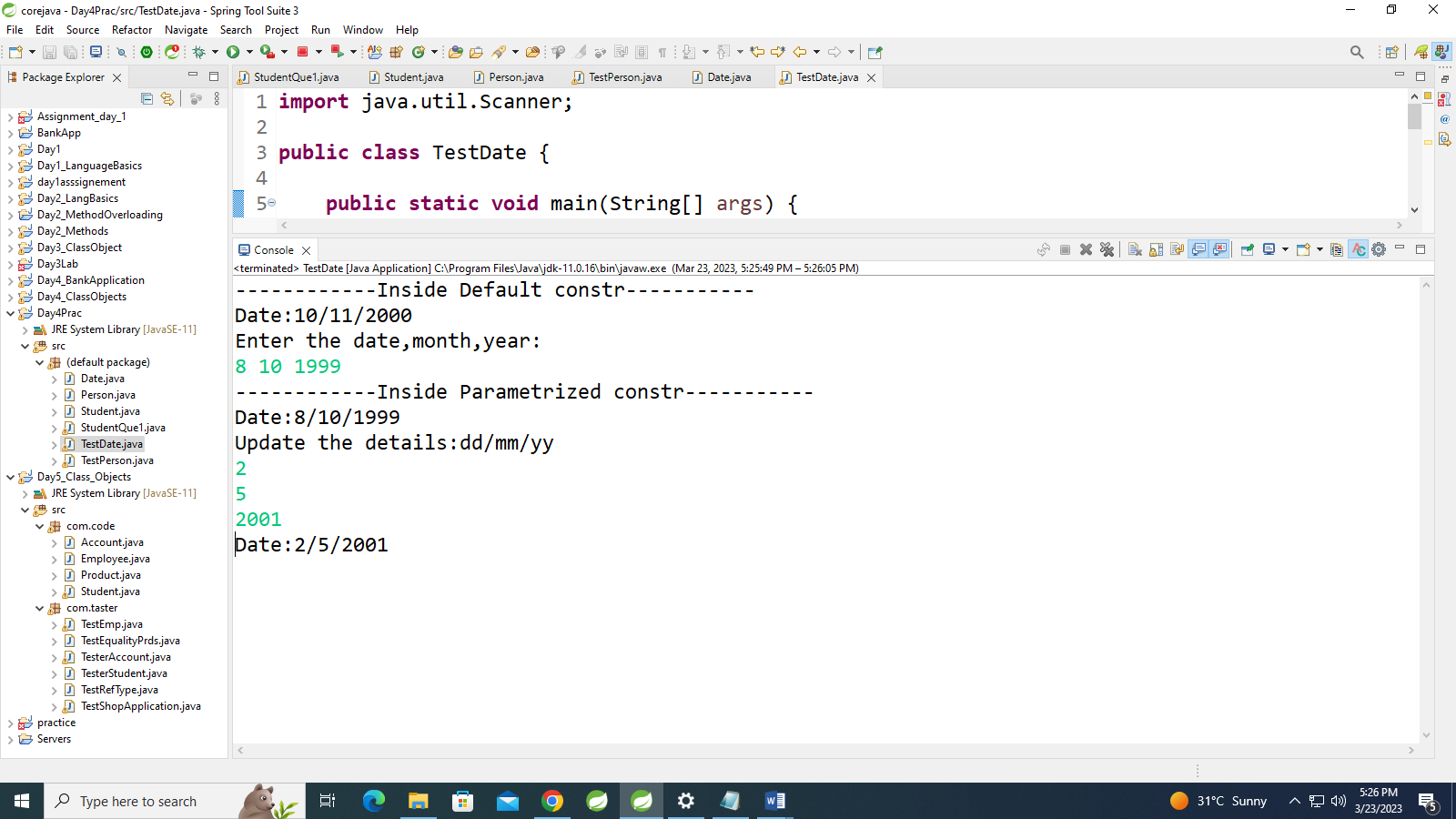
d2.setYy(s.nextInt());

System.***out***.println("Date:"+d2.getDd()+"/"+d2.getMm()+"/"+d2.getYy());

}

}

Output-



3. 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 Default and Parameterized constructors. Create

the object of this class in main method and invoke all the methods in that class.

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

**private** String name,author;

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

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

**public** Book() {

System.***out***.println("------------Inside Default constr-----------");

name = "Darker";

author = "Suruchi";

id = 101;

price =200.0;

}

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

{

System.***out***.println("------------Inside Parametrized constr-----------");

**this**.name = name;

**this**.author = author;

**this**.id = id;

**this**.price = price;

}

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

{

**this**.name=name;

}

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

{

**this**.author=author;

}

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

{

**this**.id=id;

}

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

{

**this**.price=price;

}

**public** String getName()

{

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

}

**public** String getAuthor()

{

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

}

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

{

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

}

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

{

**return** **this**.price;

}

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

{

System.***out***.println("Book id="+id+"\nName="+name+"\nAuthor="+author+"\nPrice="+price+"\n");

}

}

**import** java.util.Scanner;

**public** **class** TestBook {

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

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

Book b=**new** Book();

b.display();

Book b3=**new** Book("Java","Gosling",102,500.0);

b3.display();

System.***out***.println("Enter details to update:Name Author Id Price");

b3.setName(s.next());

b3.setAuthor(s.next());

b3.setId(s.nextInt());

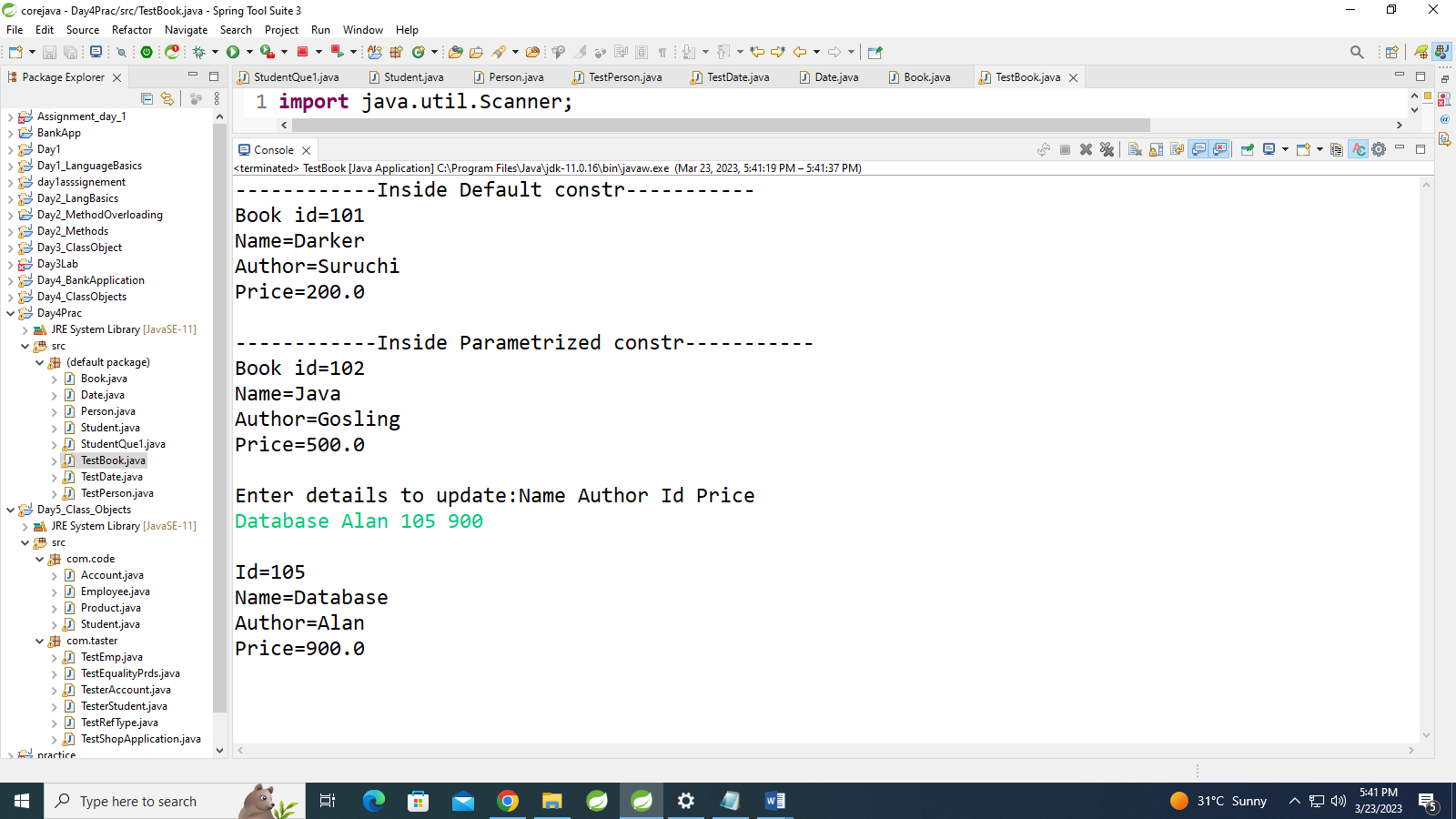
b3.setPrice(s.nextDouble());

System.***out***.println("\nId="+b3.getId()+"\nName="+b3.getName()+"\nAuthor="+b3.getAuthor()+"\nPrice="+b3.getPrice());

}

}

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.

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

**private** **int** x,y;

**public** Point() {

System.***out***.println("------------Inside Default constr-----------");

x = 3;

y =4;

}

**public** Point(**int** x, **int** y) {

System.***out***.println("------------Inside Parametrized constr-----------");

**this**.x = x;

**this**.y = y;

}

**public** **void** setx(**int** x)

{

**this**.x=x;

}

**public** **void** sety(**int** y)

{

**this**.y=y;

}

**public** **int** getx()

{

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

}

**public** **int** gety()

{

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

}

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

{

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

}

}

**import** java.util.Scanner;

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

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

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

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

p1.display();

System.***out***.println("enter values of x and y:");

Point p2=**new** Point(sc.nextInt(),sc.nextInt());

p2.display();

System.***out***.println("Enter Update values of x and y:");

p2.setx(sc.nextInt());

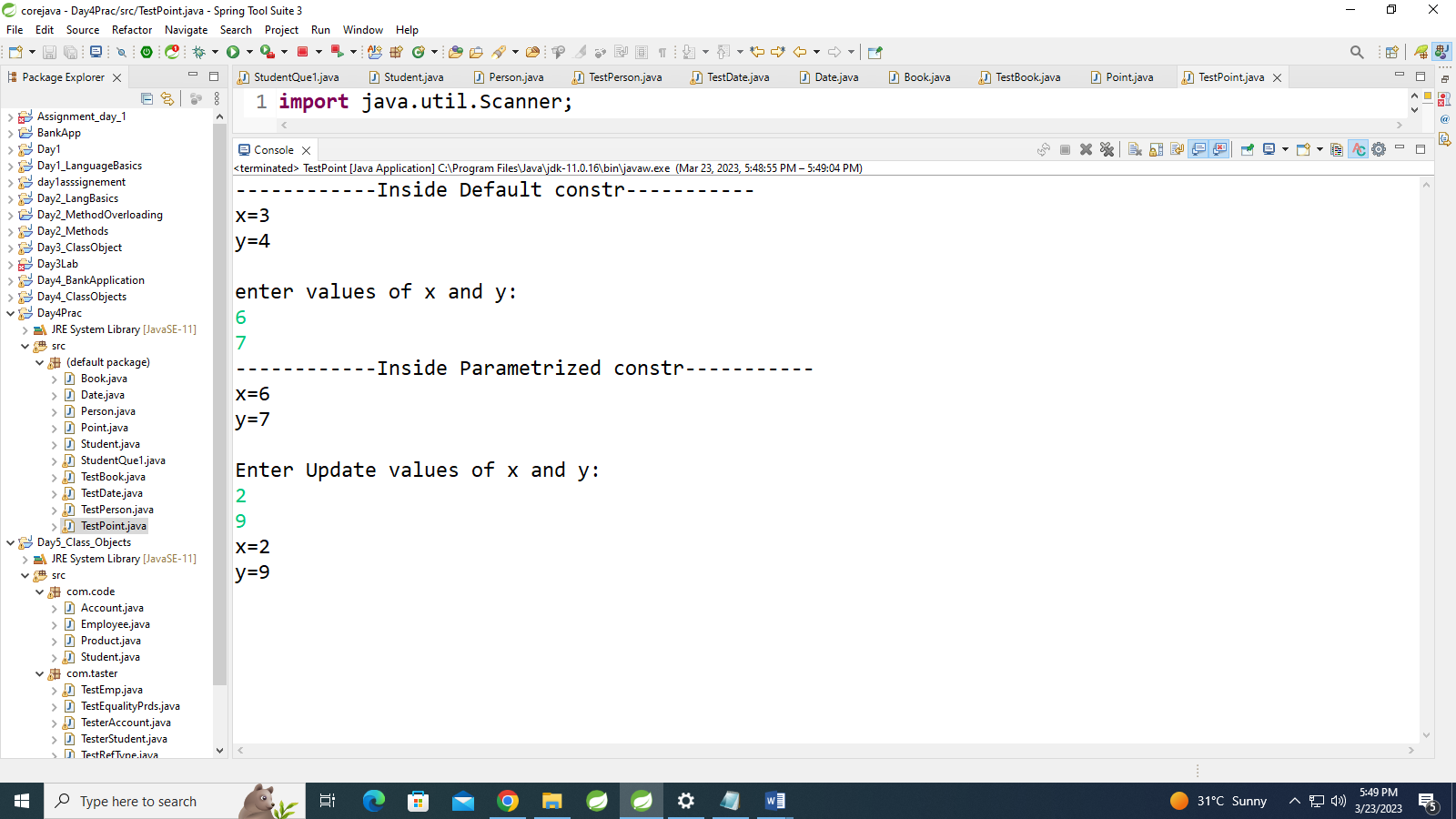
p2.sety(sc.nextInt());

System.***out***.println("x="+p2.getx()+"\ny="+p2.gety());

}

}

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.

**public** **class** ComplexNo {

**private** **int** r, i;

**public** ComplexNo() {

System.***out***.println("------------Inside Default constr-----------");

r = 10;

i = 5;

}

**public** ComplexNo(**int** r, **int** i) {

System.***out***.println("------------Inside Parametrized constr-----------");

**this**.r = r;

**this**.i = i;

}

**public** **void** setr(**int** r) {

**this**.r = r;

}

**public** **void** seti(**int** i) {

**this**.i = i;

}

**public** **int** getr() {

**return** **this**.r;

}

**public** **int** geti() {

**return** **this**.i;

}

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

System.***out***.println("complex no=" + r + "+" + i + "i\n");

}

}

**import** java.util.Scanner;

**public** **class** TestComplexNo {

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

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

ComplexNo c=**new** ComplexNo();

c.display();

System.***out***.println("Enter values of r and i:");

ComplexNo c1=**new** ComplexNo(sc.nextInt(),sc.nextInt());

c1.display();

System.***out***.println("Enter updated values of r and i:");

c1.setr(sc.nextInt());

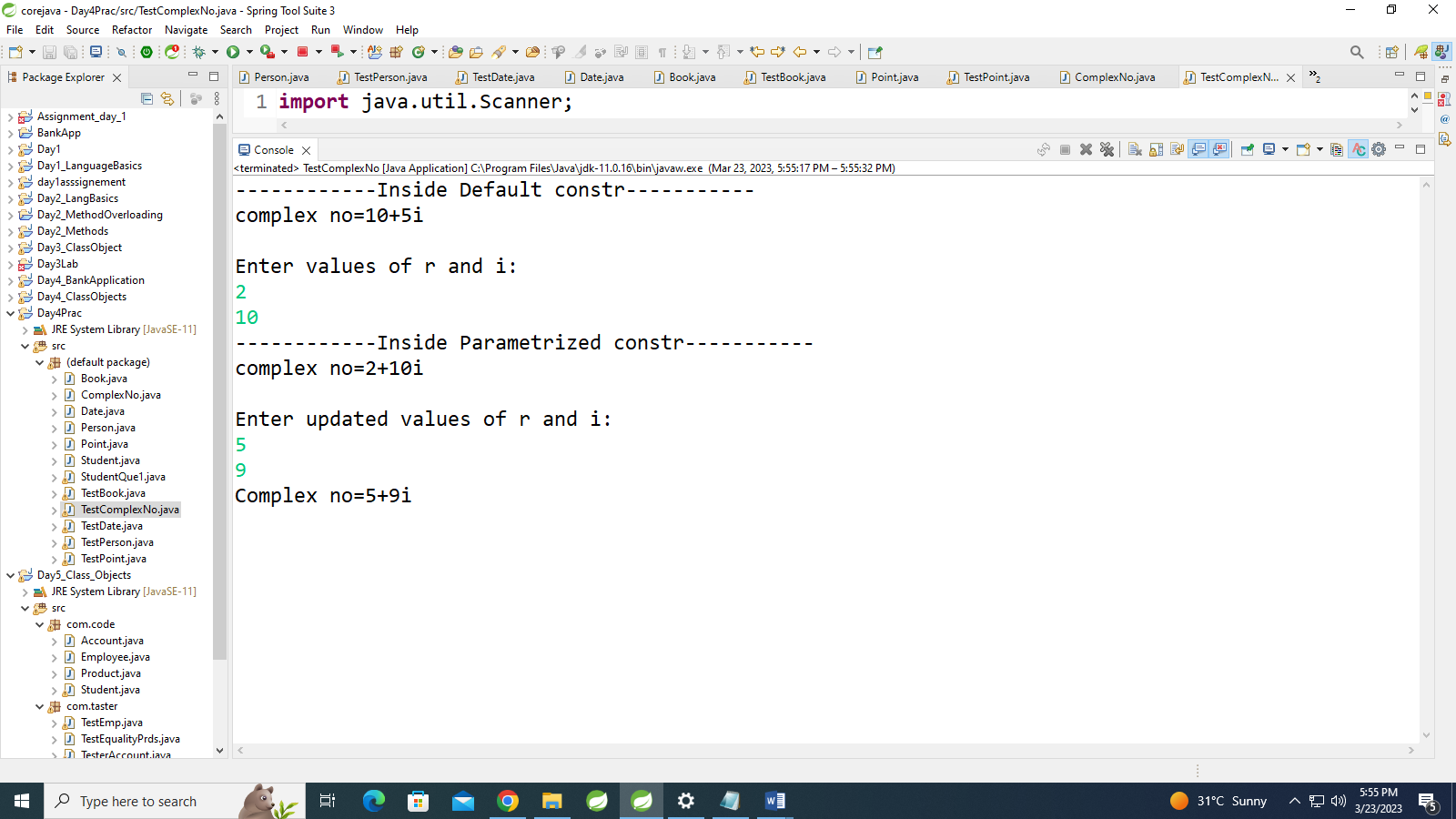
c1.seti(sc.nextInt());

System.***out***.println("Complex no="+c1.getr()+"+"+c1.geti()+"i\n");

}

}

Output-



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

Create menu drive program for bank operations..

**public** **class** Bank {

**private** **int** actid;

**private** **double** balance;

**private** String name;

**public** Bank(**int** actid,String name, **double** balance) {

**this**.actid = actid;

**this**.balance = balance;

**this**.name=name;

}

**public** String getName() {

**return** name;

}

**public** **int** getActid() {

**return** actid;

}

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

**if** (balance > amount) {

System.***out***.println("---inside withdraw----");

**this**.balance = **this**.balance - amount;

System.***out***.println("Balance Aftre Withdraw=" + **this**.balance);

} **else** {

System.***out***.println("Insufficient Amount");

}

}

**public** **double** deposite(**double** amount) {

System.***out***.println("---inside deposite----");

**this**.balance = **this**.balance + amount;

System.***out***.println("Balance Aftre deposite=" + **this**.balance);

**return** balance;

}

**public** **void** displayAccount() {

System.***out***.println("----Account Details---------");

System.***out***.println("ActId=" + actid + " Balance=" + balance);

}

**public** **void** moneyTransfer(Bank b2,**double** amount)

{

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

**this**.balance=**this**.balance-amount;

b2.balance=b2.balance+amount;

System.***out***.println("Money Transfered From :"+ **this**.getName()+" To "+b2.getName());

System.***out***.println("After Money Transafer:"+**this**.balance);

}

}

**import** java.util.Scanner;

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

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

Bank[] b = **new** Bank[10];

System.***out***.println("1:Create New Account");

System.***out***.println("2:Show All Accounts");

System.***out***.println("3:Deposite");

System.***out***.println("4:Withdraw");

System.***out***.println("5:Chk Balance");

System.***out***.println("6:Money Transfer");

System.***out***.println("7:Exit");

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

**int** choice;

**int** index = 0;

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

Bank act = **null**;

**do** {

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

choice = sc.nextInt();

**switch** (choice) {

**case** 1:

**if** (index < b.length) {

System.***out***.println("ActId Name balance");

act = **new** Bank(sc.nextInt(), sc.next(),sc.nextDouble());

b[index] = act;

index++;

} **else** {

System.***out***.println("Array Is Full");

}

**break**;

**case** 2:

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

b[i].displayAccount();

}

**break**;

**case** 3:

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

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

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

**if** (b[i].getActid() == id) {

System.***out***.println("Enter amount To Deposite");

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

b[i].deposite(amt);

flag = **true**;

**break**;

} **else** {

flag = **false**;

}

}

**if** (flag == **false**) {

System.***out***.println("Act does Not exist");

}

**break**;

**case** 4: System.***out***.println("Enter the account id:");

id=sc.nextInt();

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

{

**if**(b[i].getActid()==id)

{

System.***out***.println("Enter the amount to deposite:");

Double amt=sc.nextDouble();

b[i].deposite(amt);

flag=**true**;

**break**;

}

**else** {

flag=**false**;

}

}

**break**;

**case** 5:

**int** flg=0;

System.***out***.println("Enter the Account id:");

id=sc.nextInt();

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

{

**if**(b[i].getActid()==id)

{

flg=0;

System.***out***.println("Enter the Account id on You want to transfer:");

id=sc.nextInt();

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

{

flg=1;

**if**(b[j].getActid()==id)

{

System.***out***.println("Enter the Amount:");

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

b[i].moneyTransfer(b[j], amt);

**break**;

}

}**break**;

}

}

**if**(flg==0)

{

System.***out***.println("Account not exist");

}

**break**;

**case** 6:System.***out***.println("Exit");

**break**;

**default**:

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

**break**;

}// switch

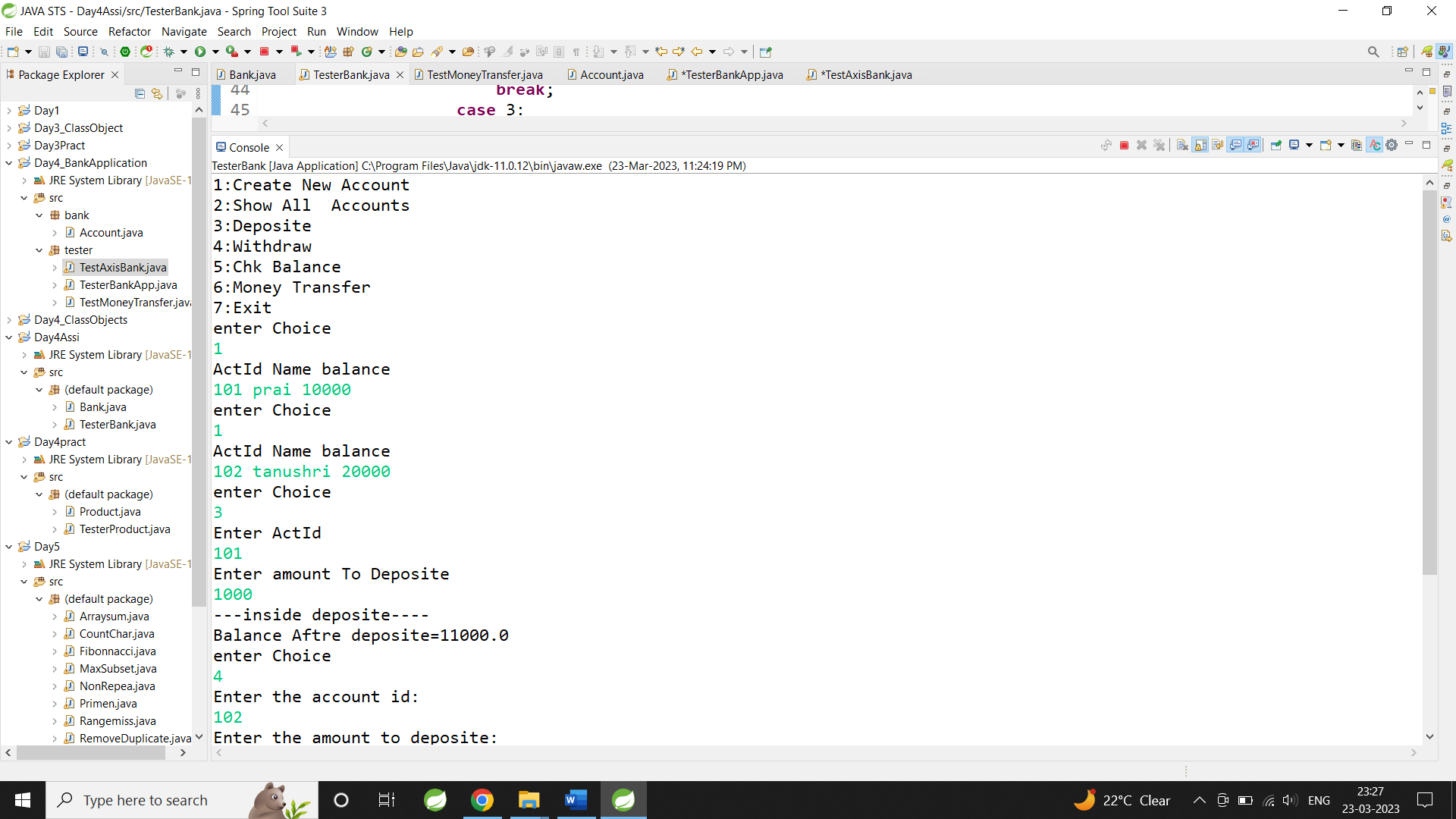
} **while** (choice !=6);

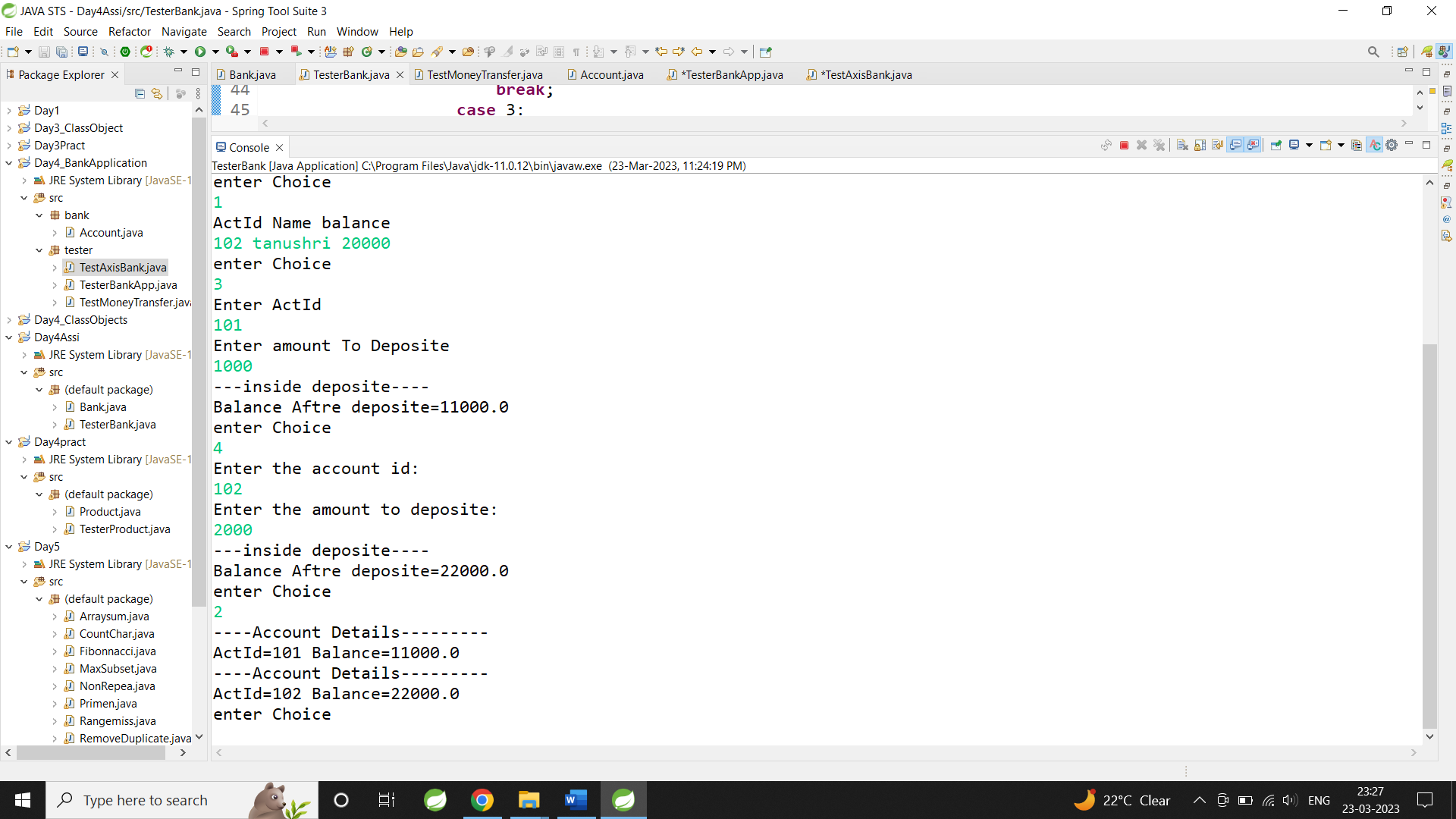
System.***out***.println("----------------end of main---------");

}

}

Output:





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

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

**int** rollno;

**static** **int** *rollid* = 0;

String name, address;

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

*rollid*++;

rollno = *rollid*;

**this**.name = name;

**this**.address = address;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** String getAddress() {

**return** address;

}

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

**this**.address = address;

}

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

System.***out***.println("Roll No: " + rollno + " Name: " + name + " Address: " + address);

}

}

**import** java.util.Scanner;

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

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

Student stu[]=**new** Student[3];

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

System.***out***.println("Enter the name and Address of student:");

**for**(**int** i=0;i<3;i++)

{

stu[i]=**new** Student(s.next(),s.next());

}

**for**(**int** i=0;i<3;i++)

{

stu[i].display();;

}

System.***out***.println("..........By using Getter...........");

System.***out***.println("Names of Student:");

**for**(**int** i=0;i<3;i++)

{

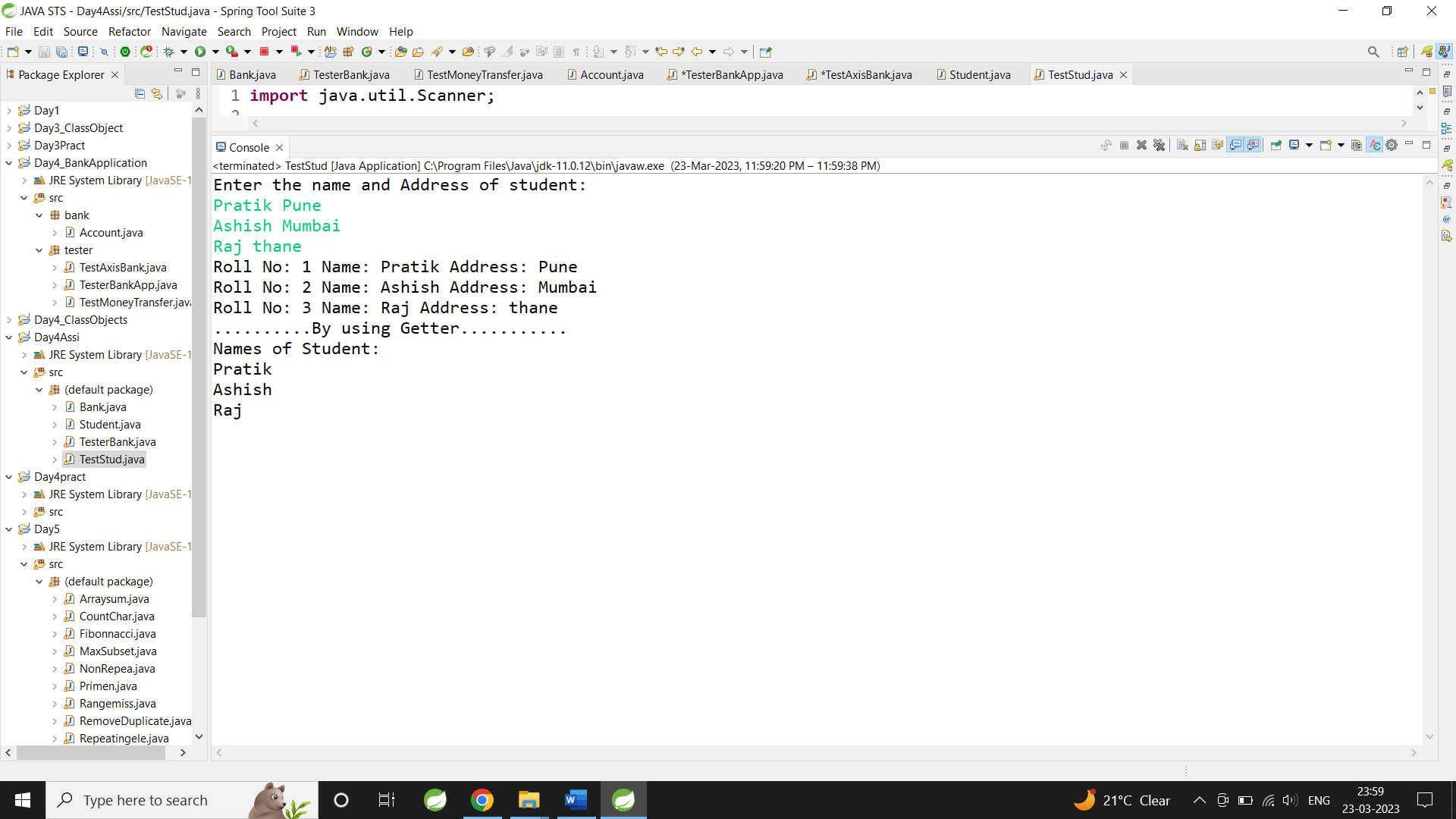
System.***out***.println(stu[i].getName());

}

}

}

Output:



7:Create diff package and add class inside that.

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

(chk default access specifier)

**package** com.code;

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

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

**public** String name ;

String address;

**static** **int** *rollid*=0;

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

*rollid*++;

rollno=*rollid*;

**this**.name = name;

**this**.address = address;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** String getAddress() {

**return** address;

}

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

**this**.address = address;

}

**public** **int** getRollno() {

**return** rollno;

}

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

System.***out***.println("Roll no :"+rollno);

System.***out***.println("Name :"+name);

System.***out***.println("Address :"+address);

}

}

**package** com.code;

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

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

**public** String name ;

String address;

**static** **int** *rollid*=0;

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

*rollid*++;

rollno=*rollid*;

**this**.name = name;

**this**.address = address;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** String getAddress() {

**return** address;

}

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

**this**.address = address;

}

**public** **int** getRollno() {

**return** rollno;

}

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

System.***out***.println("Roll no :"+rollno);

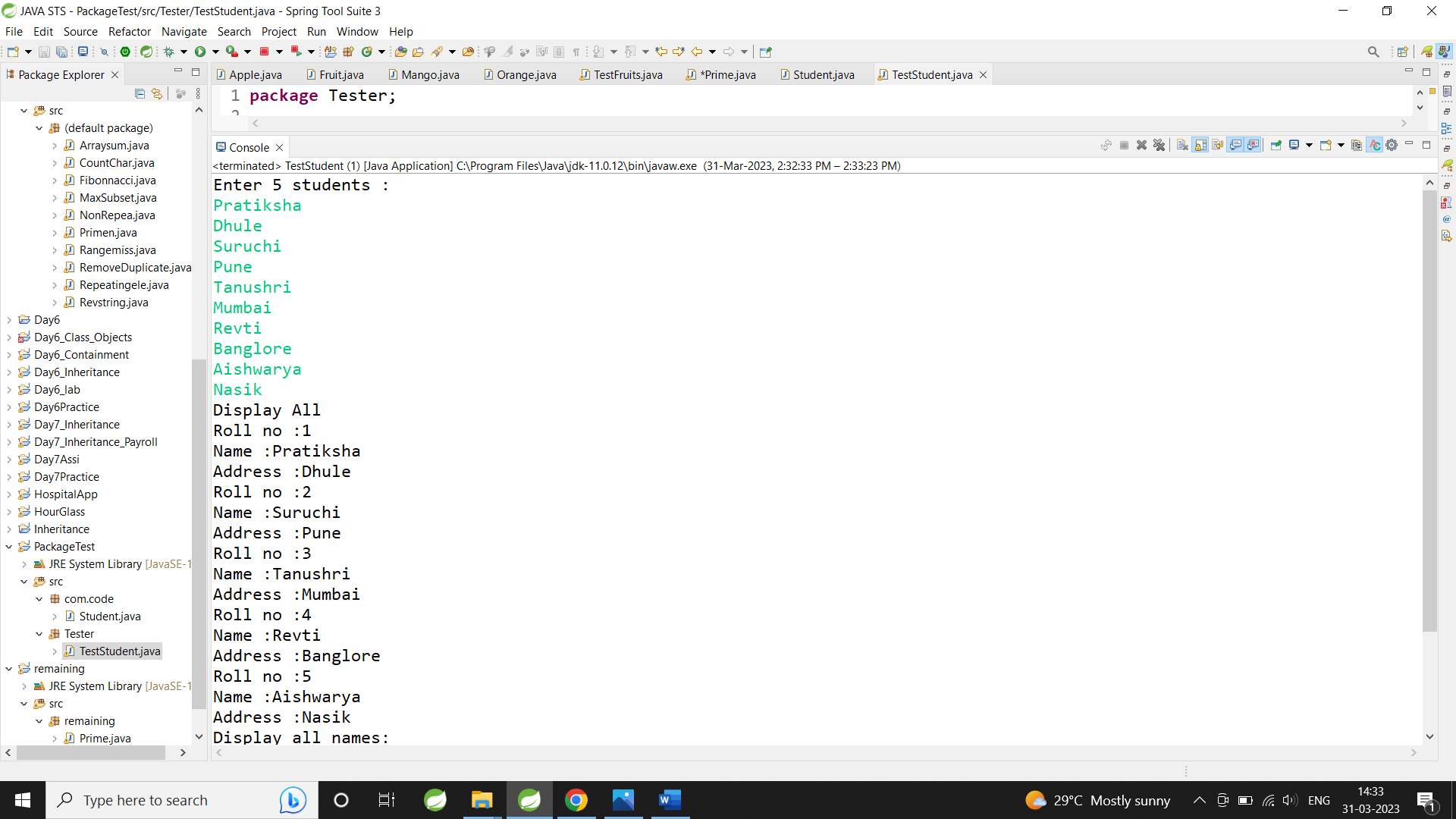
System.***out***.println("Name :"+name);

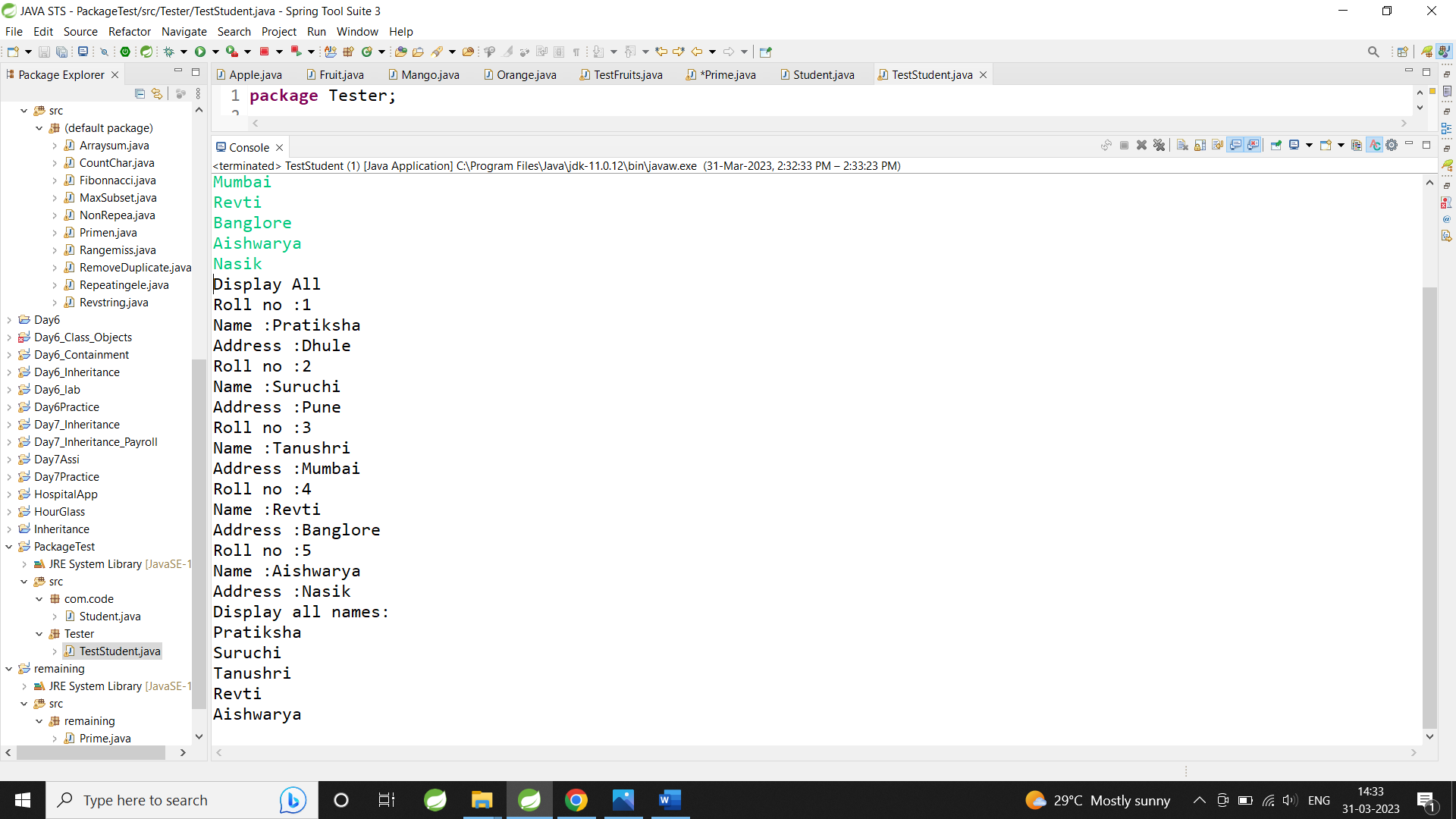
System.***out***.println("Address :"+address);

}

}

Output:





**Assignment Day 7**

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,y;

**public** Point2D(**int** x, **int** y) {

**super**();

**this**.x = x;

**this**.y = y;

}

**public** String toShow()

{

**return** "X: "+**this**.x+" Y:"+**this**.y;

}

**public** **boolean** isEqual(Point2D pr)

{

**if**(**this**.x==pr.x&&**this**.y==pr.y)

{

**return** **true**;

}

**else**

{

**return** **false**;

}

}

**public** **double** calDistance(Point2D pt)

{

**return** Math.*sqrt*(((pt.x-**this**.x)\*(pt.x-**this**.x))+((pt.y-**this**.y)\*(pt.y-**this**.y)));

}

}

package Tester;

import java.util.Scanner;

import com.cdac.Geometry.\*;

public class TestPoints {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("1.Enter the Coordinates\n2.Display the coordinates\n3.isEqual\n4.Exit");

Point2D p[]=new Point2D[10];

int ch,index=0;

do

{

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

ch=sc.nextInt();

switch(ch)

{

case 1:

if(index<p.length)

{

System.out.println("Enter the coordinates X and Y:");

p[index]=new Point2D(sc.nextInt(),sc.nextInt());

index++;

}

else

{

System.out.println("Array is full....");

}

break;

case 2:

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

{

String s=p[i].toShow();

System.out.println(s);

}

break;

case 3:

System.out.println("Enter the coordinates(1 Indices) X and Y:");

Point2D p1=new Point2D(sc.nextInt(),sc.nextInt());

System.out.println("Enter the coordinates(2 Indices) X and Y:");

Point2D p2=new Point2D(sc.nextInt(),sc.nextInt());

boolean a=p1.isEqual(p2);

if(a==true)

{

System.out.println(" POints are same ");

}

else

{

double c=p1.calDistance(p2);

System.out.println("Distance betweetn two point is:"+c);

}

break;

case 4:

break;

default:

break;

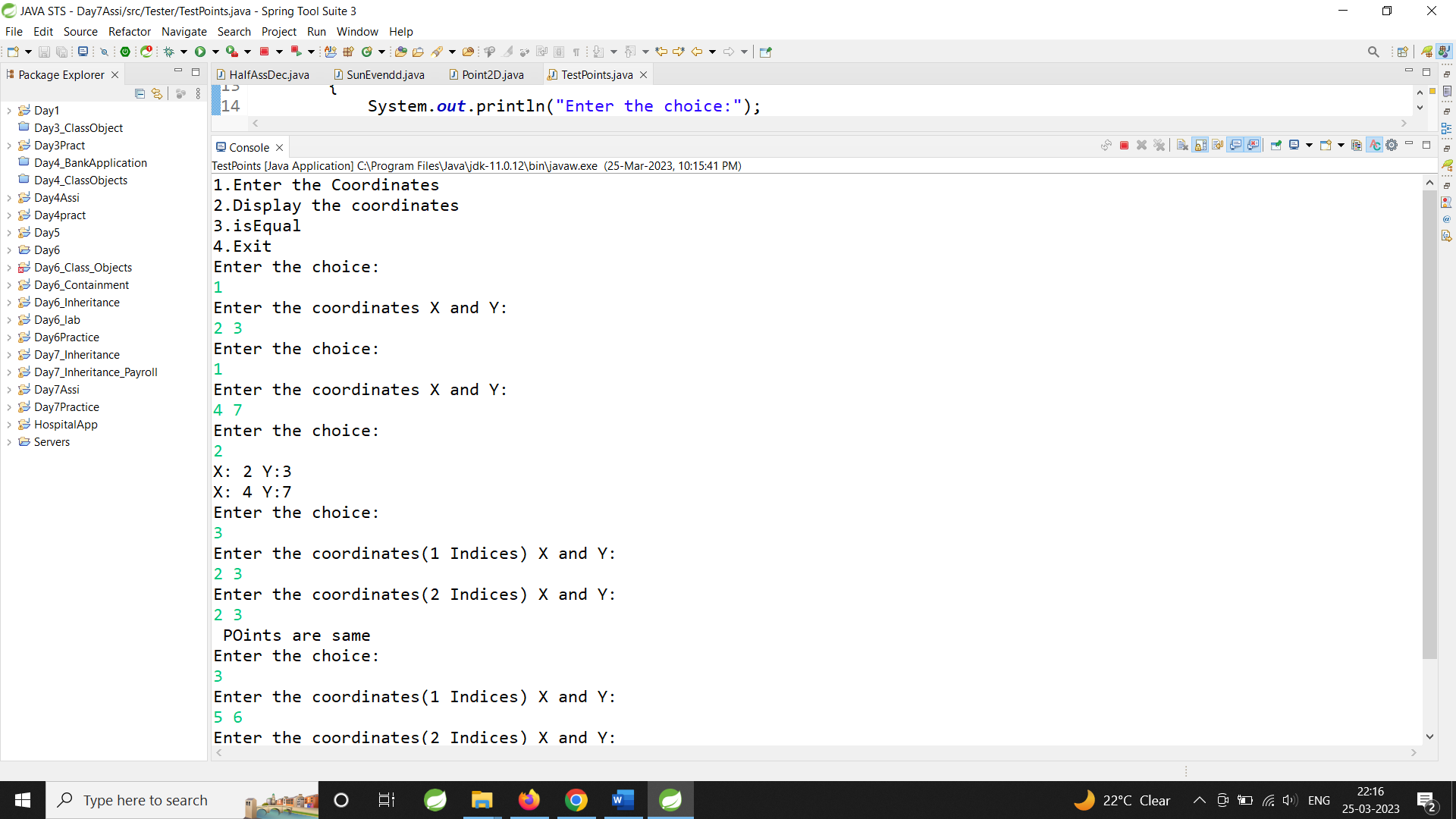
}

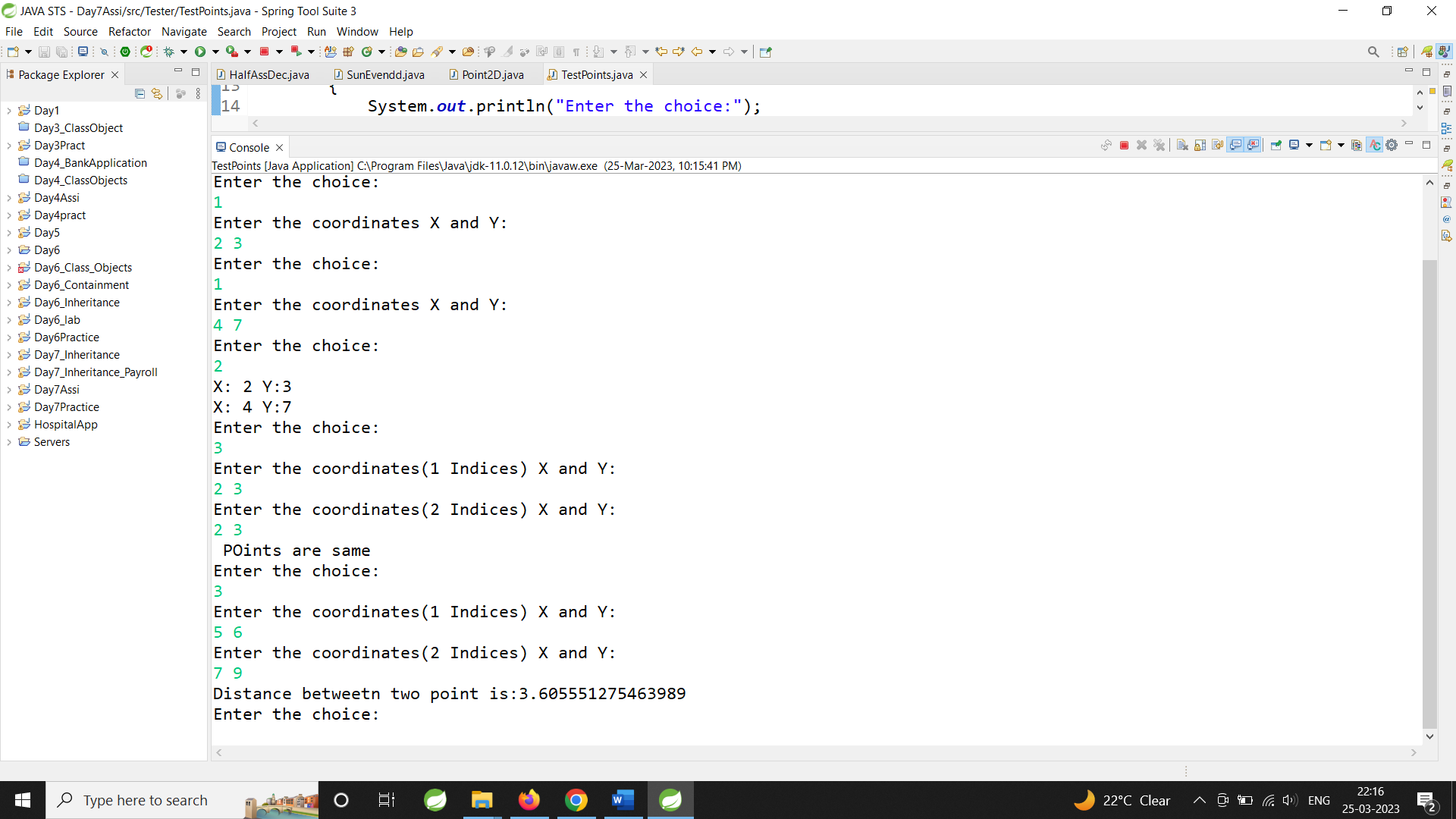
}while(ch!=4);

}

}

Output:





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** com.app.fruits;

**public** **class** Fruits {

**protected** String color;

**protected** String name;

**protected** **double** weight;

**protected** **boolean** fresh;

**public** Fruits()

{

color="yellow";

name="manngo";

weight=50.0;

fresh=**true**;

}

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

**super**();

**this**.color = color;

**this**.name = name;

**this**.weight = weight;

**this**.fresh = fresh;

}

@Override

**public** String toString() {

**return** color + " " + name + " " + weight ;

}

**public** String getColor() {

**return** color;

}

**public** String getName() {

**return** name;

}

**public** **double** getWeight() {

**return** weight;

}

**public** **boolean** getFresh()

{

**return** fresh;

}

**public** **void** setFresh(**boolean** fresh)

{

**this**.fresh=fresh;

}

**public** String taste()

{

**return** "no specific taste";

}

}

**package** com.app.fruits;

**public** **class** Apple **extends** Fruits {

**public** Apple()

{

color="red";

name="apple";

weight=100.0;

fresh=**true**;

}

**public** Apple(String c,String n,**double** w,Boolean f)

{

**super**(c,n,w,f);

}

**public** **void** jam()

{

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

}

**public** String taste()

{

**return** "sweet n sour";

}

**public** String toString()

{

**return** **super**.color+" "+**super**.name+" "+**super**.weight+" "+**super**.fresh+"\n";

}

}

**package** com.app.fruits;

**public** **class** Mango **extends** Fruits{

**public** Mango()

{

color="yellow";

name="manngo";

weight=50.0;

fresh=**true**;

}

**public** Mango(String c,String n,**double** w,Boolean f)

{

**super**(c,n,w,f);

}

**public** **void** pulp()

{

System.***out***.println(**super**.toString()+" creating pulp");

}

**public** String taste()

{

**return** "sweet";

}

**public** String toString()

{

**return** **super**.color+" "+**super**.name+" "+**super**.weight+" "+**super**.fresh+"\n";

}

}

**package** com.app.fruits;

**public** **class** Orange **extends** Fruits {

**public** Orange() {

color = "orange";

name = "orange";

weight = 150.0;

fresh = **true**;

}

**public** Orange(String c, String n, **double** w, Boolean f) {

**super**(c, n, w, f);

}

**public** **void** juice() {

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

}

**public** String taste() {

**return** "sour";

}

**public** **boolean** getFresh() {

**return** **super**.fresh;

}

**public** **void** setFresh(**boolean** fresh) {

**this**.fresh = fresh;

}

**public** String toString() {

**return** **super**.color + " " + **super**.name + " " + **super**.weight + " " + **super**.fresh + "\n";

}

}

package tester;

import java.util.Scanner;

import com.app.fruits.Fruits;

import com.app.fruits.Mango;

import com.app.fruits.\*;

public class FruitBasket {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter value of n:");

int n = sc.nextInt();

Fruits arr[] = new Fruits[n];

Fruits f = new Fruits();

int index = 0;

int ch;

System.out.println("\n1. Add Mango \n2. Add Orange\n3.Add Apple\n4.Display names of "

+ "fruits\n5.Display name color weight taste of all fresh fruits\n6.Mark fruit"

+ " as a stale(not fresh)\n7.Mark all sour fruits stale\n8.Invoke fruits "

+ "Specific functionality\n10.Exit\n");

do {

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

ch = sc.nextInt();

switch (ch) {

case 1:

if (index < arr.length) {

System.out.println("Enter color name weight fresh:");

Mango m = new Mango(sc.next(), sc.next(), sc.nextDouble(), sc.nextBoolean()); // upcasting

arr[index] = m;

index++;

} else {

System.out.println("Array is full");

}

break;

case 2:

if (index < arr.length) {

System.out.println("Enter color name weight fresh:");

Orange o = new Orange(sc.next(), sc.next(), sc.nextDouble(), sc.nextBoolean()); // upcasting

arr[index] = o;

index++;

}

break;

case 3:

if (index < arr.length) {

System.out.println("Enter color name weight fresh:");

f = new Apple(sc.next(), sc.next(), sc.nextDouble(), sc.nextBoolean()); // upcasting

arr[index] = f;

index++;

}

break;

case 4:// displays names of all fruit

System.out.println("Displays of all names:");

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

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

}

break;

case 5:

System.out.println("List of all fruits");

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

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

}

break;

case 6:

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

if ((arr[i].getFresh()) == true) {

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

} else {

System.out.println("it is stale");

}

}

break;

case 7:

// 7. Mark all sour fruits stale

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

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

if ((arr[i].taste()).equals("sour")) {

arr[i].setFresh(false);

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

} else {

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

}

}

break;

case 8:

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

System.out.println("enter index:1.mango\n 2.Orange\n3.Apple");

int in = sc.nextInt();

if (in == 1) {

Mango m = new Mango();

m.pulp();

} else if (in == 2) {

Orange o1 = new Orange();

o1.juice();

} else if (in == 3) {

Apple a = new Apple();

a.jam();

} else {

System.out.println("enter valid choics...");

}

break;

case 9:

break;

default:

break;

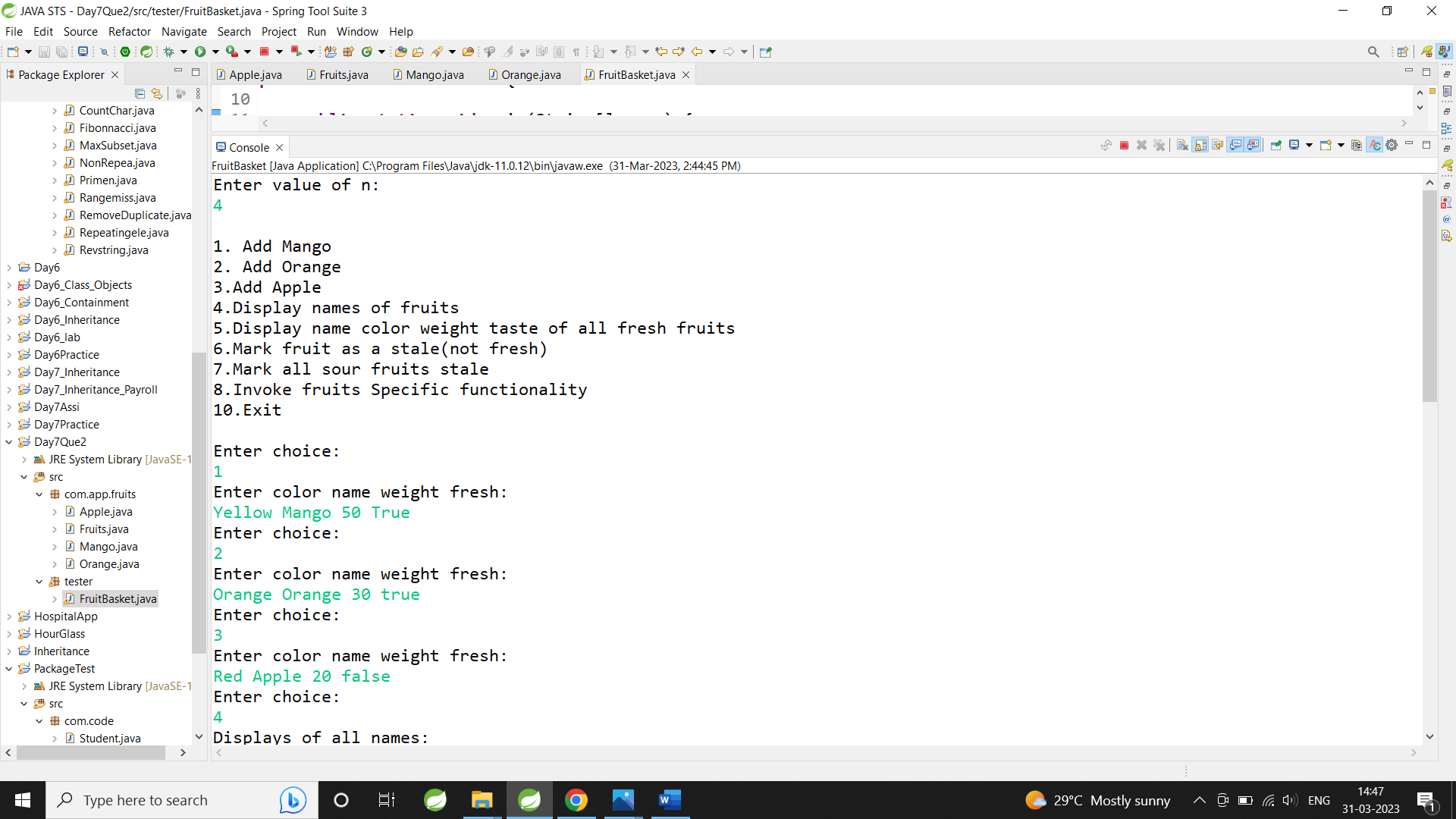
}

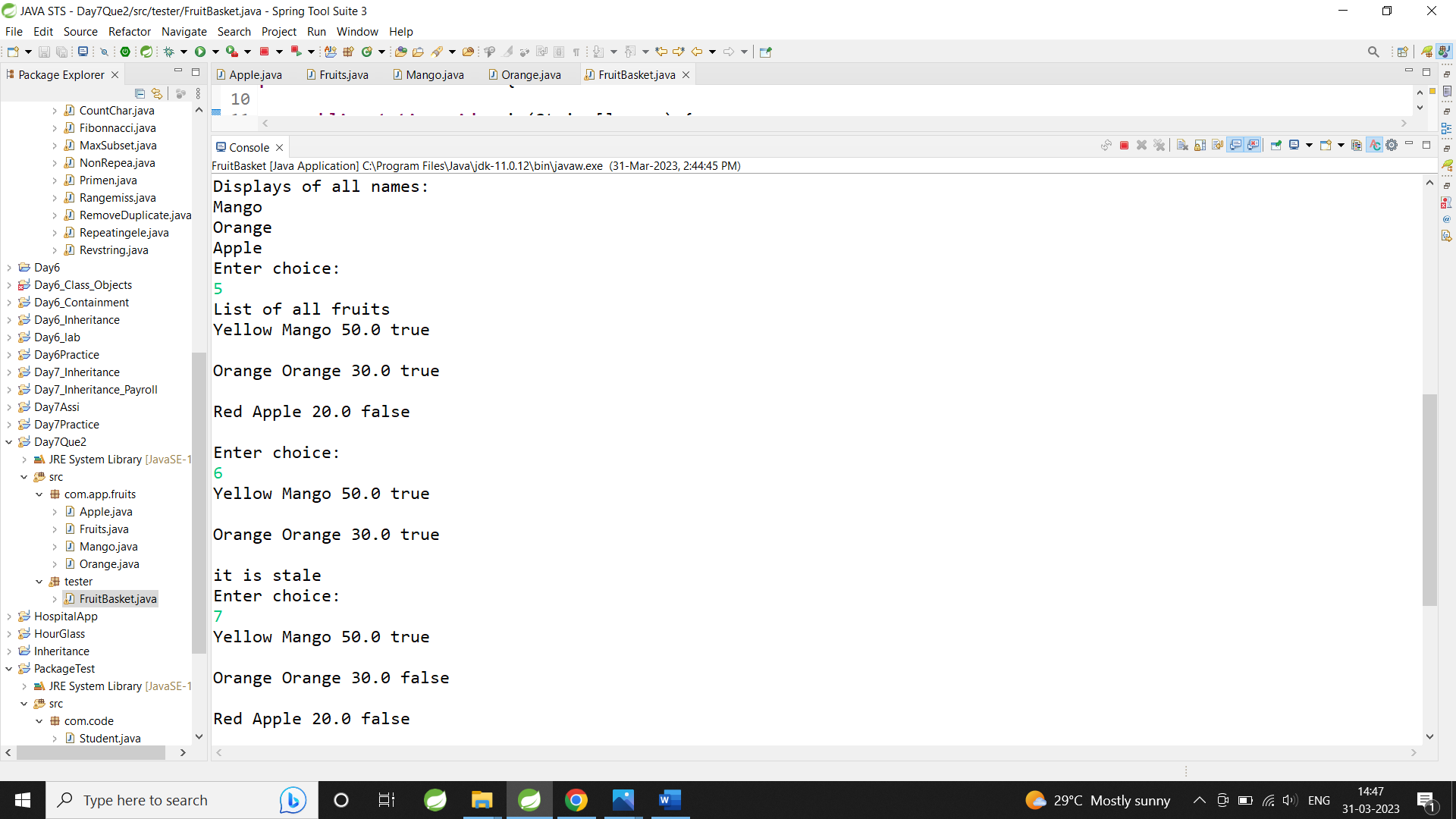
} while (ch != 9);

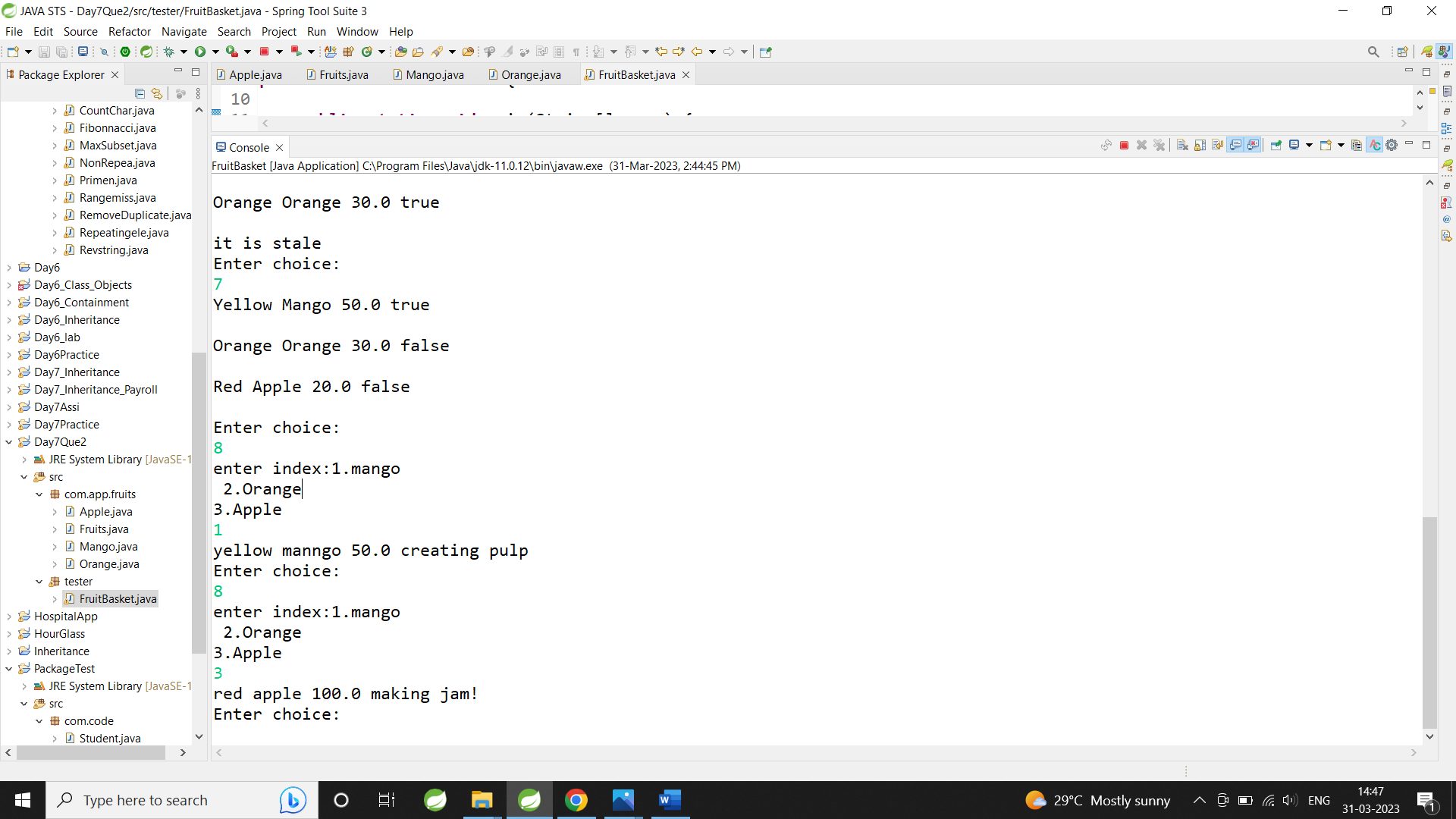
}

}

Output:







TAsk 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

Write TestOrganization in "tester" package.

Create suitable array to store organization details.

Provide following options

1. Hire Manager

i/p : manager details , except id

2. Hire Worker

i/p : worker details , except id

3. Display information of all employees(toString) including net salary(computeNetSalary) using single for-each loop.

4. Update basic salary

i/p : emp id , salary increment

In case of invalid emp id , either : display error message directly

OR (still better)

throw custom exception n handle it in centralized manner

(This can be added later!)

10 Exit

**package** com.app.emp;

**public** **class** Employee {

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

**private** **int** deptid, empid;

**private** String fisrstName, lastName;

**private** **double** basicSal;

**static** {

*id* = 100;

}

**public** Employee(**int** deptid, String fisrstName, String lastName, **double** basicSal) {

**super**();

*id*++;

**this**.empid = *id*;

**this**.deptid = deptid;

**this**.fisrstName = fisrstName;

**this**.lastName = lastName;

**this**.basicSal = basicSal;

}

**public** **double** getBasicSal() {

**return** basicSal;

}

**public** **void** setBasicSal(**double** basicSal) {

**this**.basicSal = basicSal;

}

@Override

**public** String toString() {

**return** "Id: " + empid + " Department Id: " + deptid + " fisrstName=" + fisrstName + " lastName=" + lastName

+ " basicSal=" + basicSal;

}

// = basic+(hoursWorked\*hourlyRate)

**public** **double** netSal() {

**return** basicSal;

}

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

**return** empid;

}

}

**package** com.app.emp;

**public** **class** Manager **extends** Employee {

**private** **int** perfBonus;

**public** Manager(**int** deptid, String fisrstName, String lastName, **double** basicSal, **int** perfBonus) {

**super**(deptid, fisrstName, lastName, basicSal);

**this**.perfBonus = perfBonus;

}

**public** **double** netSal() {

**return** (**super**.netSal() + perfBonus);

}

**public** **int** getPerfBonus() {

**return** perfBonus;

}

@Override

**public** String toString() {

**double** a = netSal();

**return** **super**.toString() + " Net sal: " + a;

}

}

**package** com.app.emp;

**public** **class** Worker **extends** Employee {

**private** **int** hourWork, hourRate;

**public** Worker(**int** deptid, String fisrstName, String lastName, **double** basicSal, **int** hourWork, **int** hourRate) {

**super**(deptid, fisrstName, lastName, basicSal);

**this**.hourRate = hourRate;

**this**.hourWork = hourWork;

}

**public** **double** netSal() {

**return** (**super**.netSal() + (hourWork \* hourRate));

}

**public** **int** getHourRate() {

**return** hourRate;

}

@Override

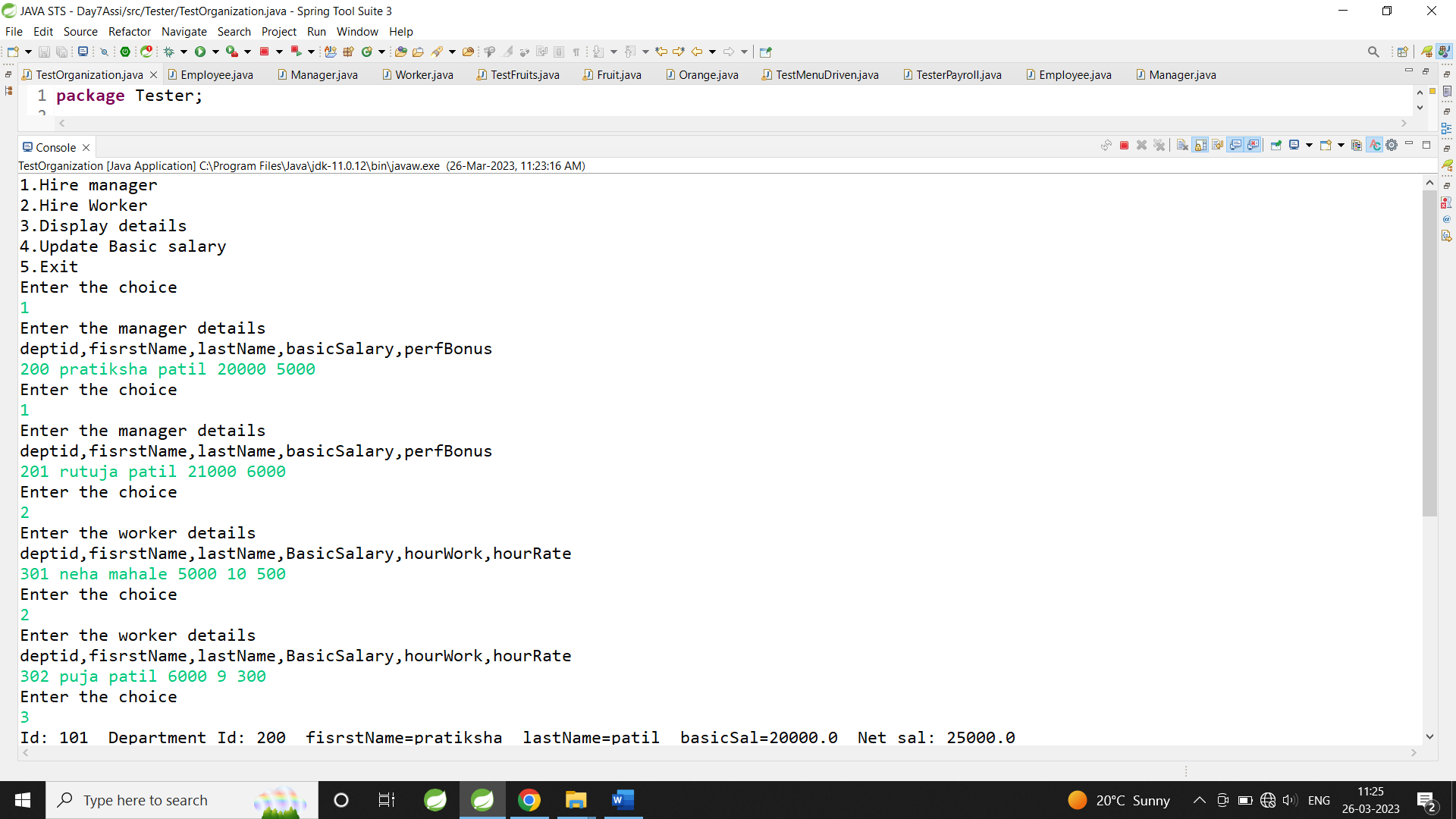
**public** String toString() {

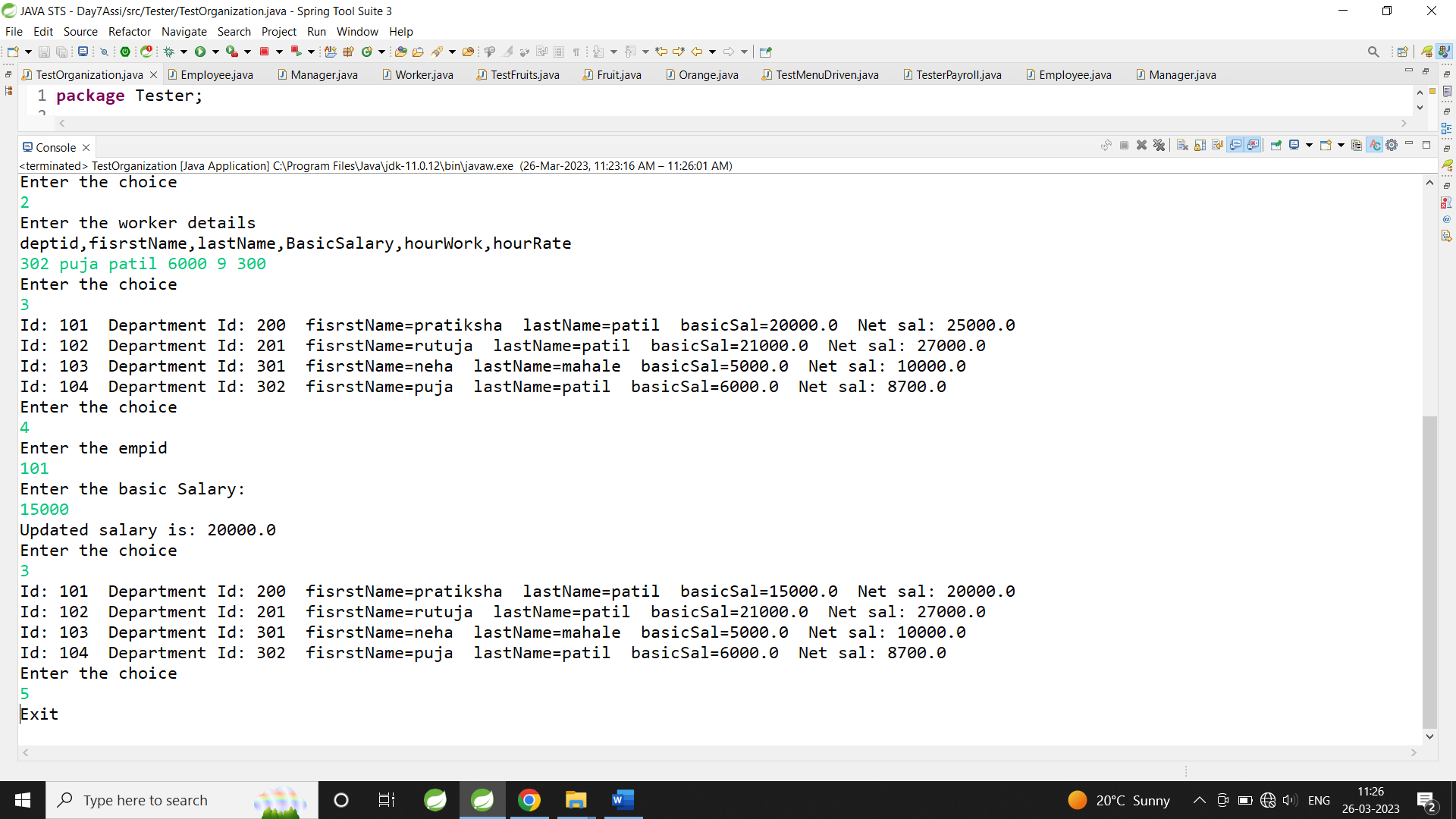
**double** a = netSal();

**return** **super**.toString() + " Net sal: " + a;

}

Output:





**Assignment Day 8**

**: Solve assignment (to understand abstraction)**

**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**

**abstract** **class** Shape {

**int** x, y;

**abstract** **public** **double** area();

**public** String toString() {

**return** x + " " + y;

}

}

**public** **class** Circle **extends** Shape {

**int** rad, x, y;

**public** Circle(**int** x, **int** y, **int** rad) {

**this**.x = x;

**this**.y = y;

**this**.rad = rad;

}

@Override

**public** **double** area() {

**return** 3.14 \* rad \* rad;

}

**public** String toString() {

**return** " X: " + x + " Y: " + y + " Area: " + 3.14 \* rad \* rad + "\n";

}

}

**public** **class** Square **extends** Shape {

**int** x, y, side;

**public** Square(**int** x, **int** y, **int** side) {

**this**.x = x;

**this**.y = y;

**this**.side = side;

}

@Override

**public** **double** area() {

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

**return** side \* side;

}

**public** String toString() {

**return** " X: " + x + " Y: " + y + " Area: " + side \* side + "\n";

}

}

**public** **class** Reactangle **extends** Shape {

**int** w, h;

**public** Reactangle(**int** x, **int** y, **int** w, **int** h) {

**this**.x = x;

**this**.y = y;

**this**.w = w;

**this**.h = h;

}

@Override

**public** **double** area() {

**return** w \* h;

}

**public** String toString() {

**return** " X: " + x + " Y: " + y + " Area: " + w \* h + "\n";

}

}

**import** java.util.Random;

**public** **class** ShapeFactory {

**static** **int** generateShape() {

Random rand = **new** Random();

**int** no = rand.nextInt(3);

**return** no;

}

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

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

**int** n = *generateShape*();

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

System.***out***.print(" Area of Circle:");

Shape s = **new** Circle(10, 20, 15);

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

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

} **else** **if** (n == 1) {

System.***out***.print(" Area of Square:");

Shape s1 = **new** Square(10, 20, 15);

s1.area();

System.***out***.println(s1.area());

System.***out***.println(s1);

} **else** {

System.***out***.print(" Area of Rectangle:");

Shape s2 = **new** Reactangle(5, 6, 7, 8);

s2.area();

System.***out***.println(s2.area());

System.***out***.println(s2);

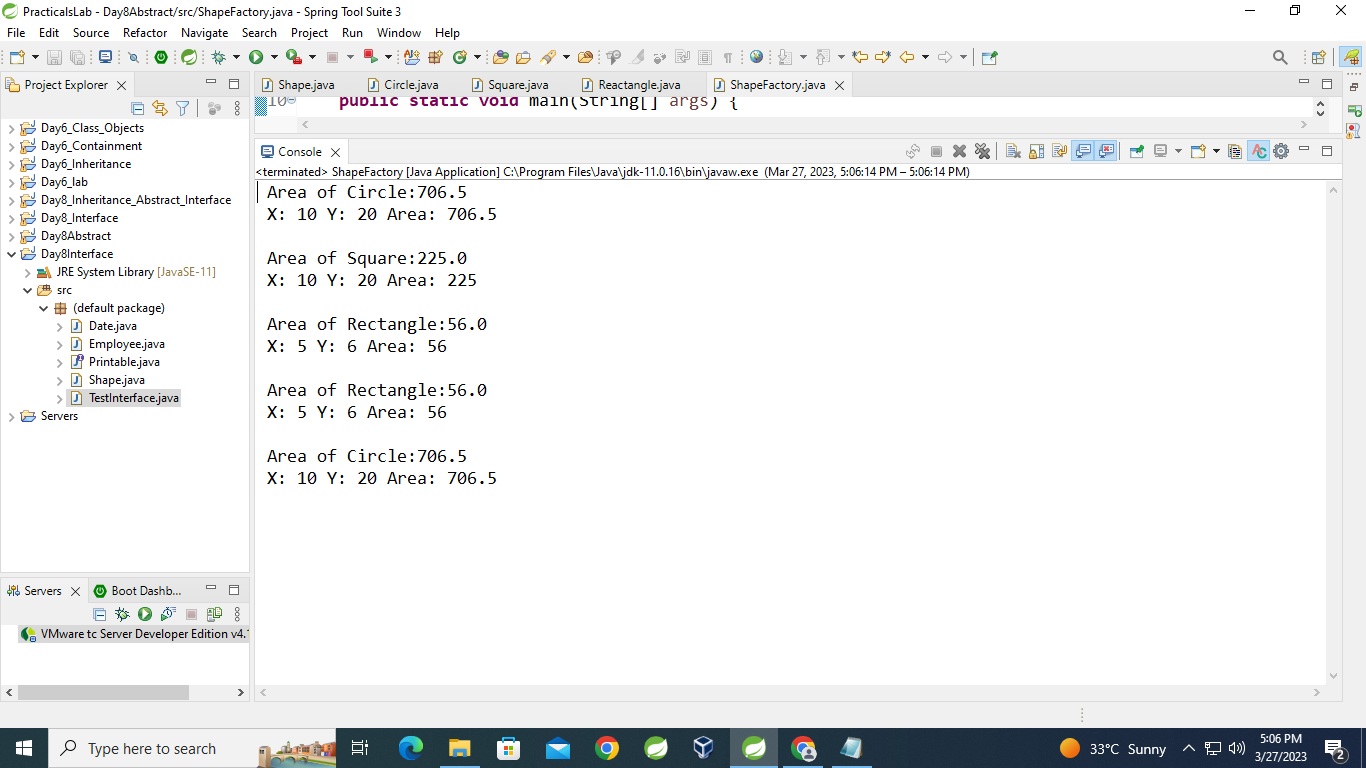
}

}

}

}

Output

****

**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.**

**public** **interface** Printable {

**void** print();

}

**public** **class** Employee **implements** Printable{

@Override

**public** **void** print() {

System.***out***.println("Print method inside Employee");

}

}

**public** **class** Shape **implements** Printable{

@Override

**public** **void** print() {

System.***out***.println("Print method inside Shape");

}

}

**public** **class** Date **implements** Printable{

@Override

**public** **void** print() {

System.***out***.println("Print method inside Date");

}

}

**public** **class** TestInterface {

**public** **static** **void** display(Printable p) {

p.print();

}

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

Printable pt = **new** Employee();

Printable pt1 = **new** Date();

Printable pt2 = **new** Shape();

pt.print();

pt1.print();

pt2.print();

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

*display*(pt);

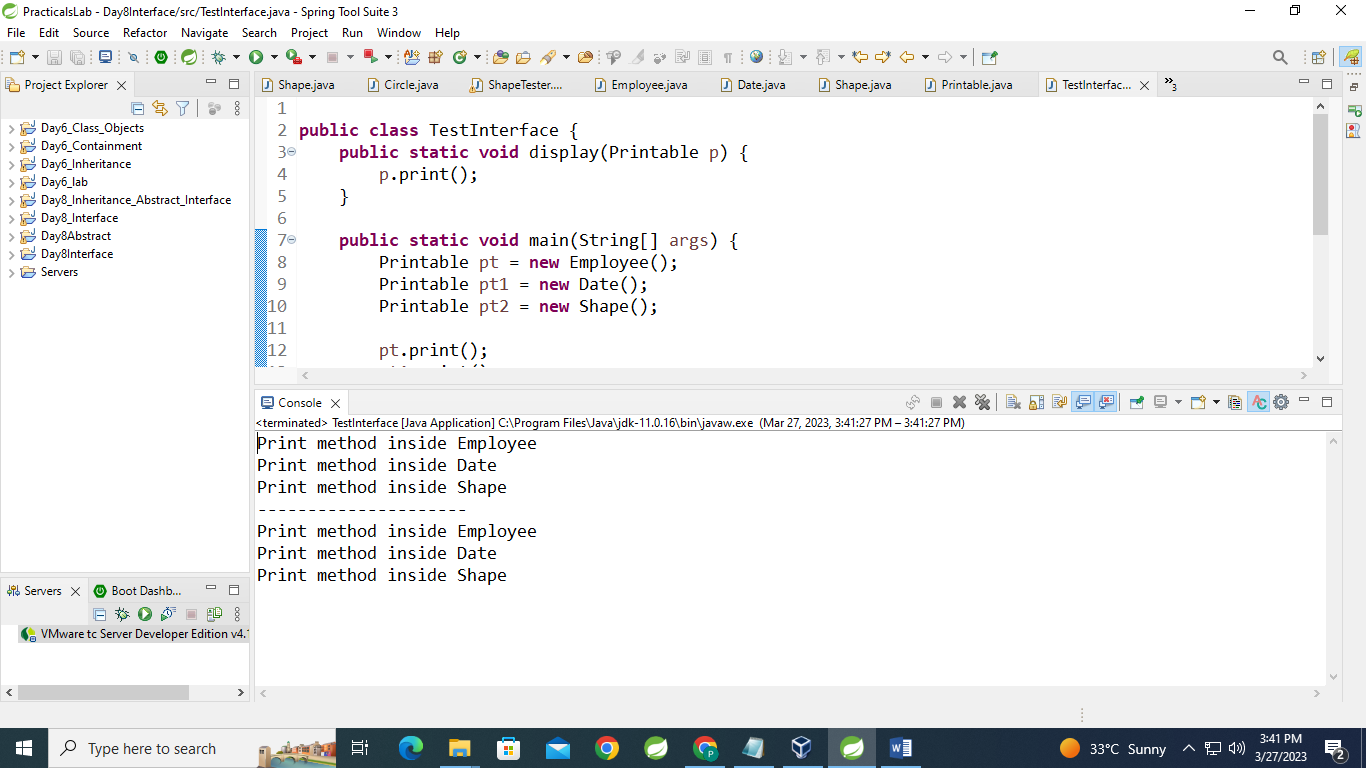
*display*(pt1);

*display*(pt2);

}

}

**Output:**

****