

# **OBJECT ORIENTED PROGRAMMING LAB RECORD**

**SRUTHY VARGHESE**

**RMCA B 26**

### Program no:1

#### Product

Aim: Define a class 'product' with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

Program:

```
public class Product
{
    int pcode;
    String pname;
    int price;

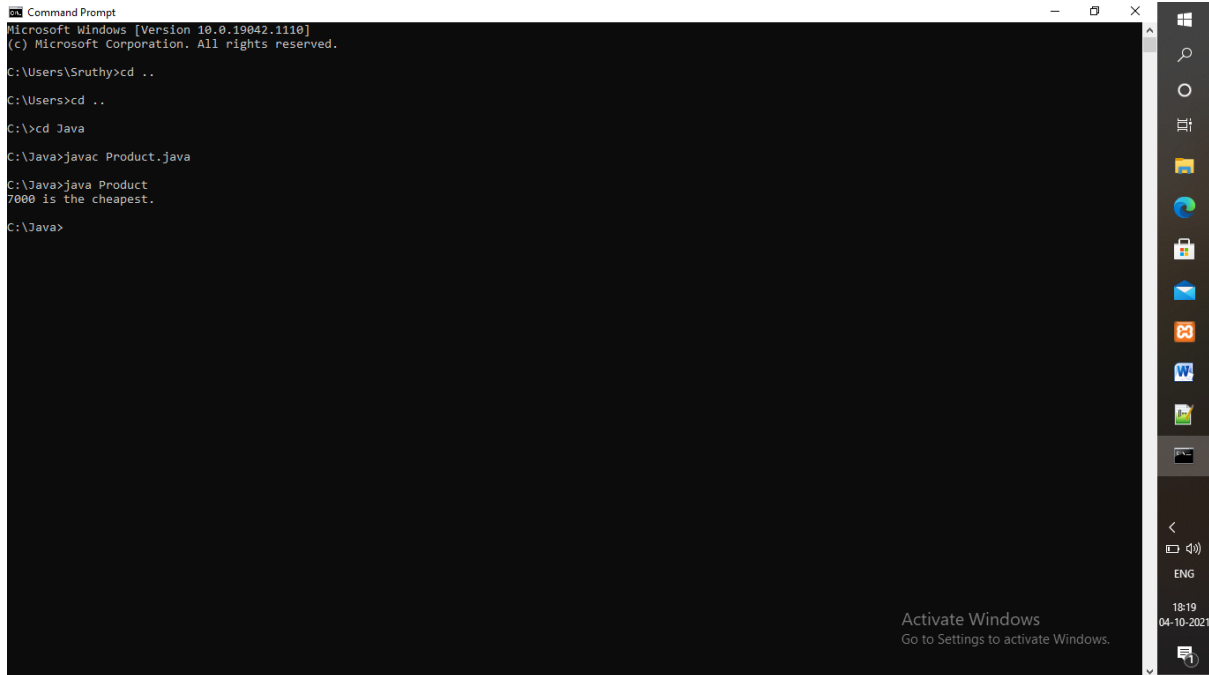
    public static void main(String[] args) {
        int smallest;

        Product p1 = new Product();
        Product p2 = new Product();
        Product p3 = new Product();

        p1.pcode=1001;
        p1.pname="RAM";
        p1.price=7000;
        p2.pcode=1002;
        p2.pname="Processor";
```

```
p2.price=37000;
p3.pcode=1001;
p3.pname="SSD";
p3.price=16700;
if(p1.price<p2.price) {
if(p3.price<p1.price) {
smallest = p3.price;
} else {
smallest = p1.price;
}
} else {
if(p2.price<p3.price) {
smallest = p2.price;
} else {
smallest = p3.price;
}
}
System.out.println(smallest + " is the cheapest.");
}
}
```

## Output:



```
Command Prompt
Microsoft Windows [Version 10.0.19042.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sruthy>cd ..
C:\Users>cd ..
C:\>cd Java
C:\Java>javac Product.java
C:\Java>java Product
7000 is the cheapest.
C:\Java>
```

## Program no: 2

### Matrix addition1

Aim: Read 2 matrices from the console and perform matrix addition.

Program:

```
import java.util.*;

class matrixadd1{

public static void main(String[] args)

{

int row,col,i,j;
```

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

System.out.print("enter the no of rows:");

row=sc.nextInt();

System.out.print("enter the no of columns:");
col=sc.nextInt(); int mat1[][]=new

int[row][col];

int mat2[][]=new int[row][col];

int mat3[][]=new int[row][col];

System.out.print("enter the elements of matrix1 :");

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

mat1[i][j]=sc.nextInt();

}

System.out.println();

}

System.out.print("enter the elements of matrix2 :");

for(i=0;i<row;i++)

{
```

```
for(j=0;j<col;j++)
{
mat2[i][j]=sc.nextInt();
}
System.out.println();
}
for(i=0;i<row;i++)
{
for(j=0;j<col;j++)
{
mat3[i][j]=mat1[i][j]+mat2[i][j];
}
}
System.out.print("sum of matrix :\n");
for(i=0;i<row;i++)
{
for(j=0;j<col;j++)
{
System.out.print(mat3[i][j]+"\\t");
}
}
```

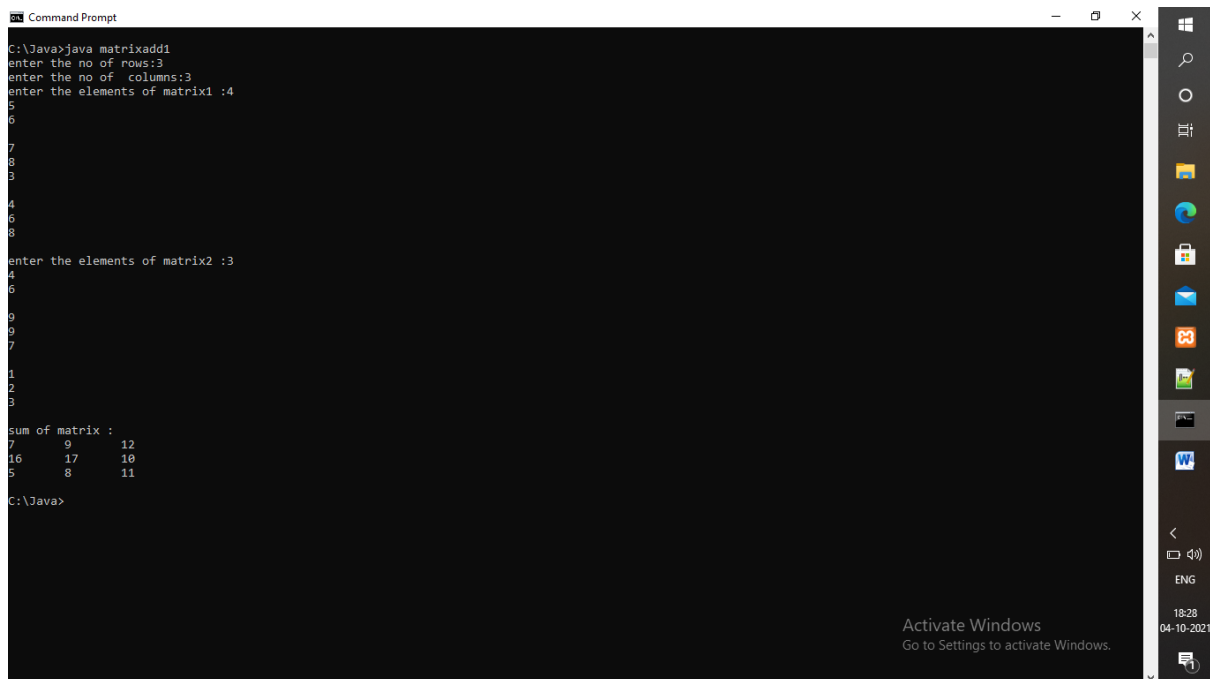
```
System.out.println();
```

```
}
```

```
}
```

```
}
```

Output:



```
Command Prompt
C:\Java>java matrixadd1
enter the no of rows:3
enter the no of columns:3
enter the elements of matrix1 :4
5
6
7
8
3
4
6
8
enter the elements of matrix2 :3
4
6
9
9
7
1
2
3
sum of matrix :
7    9    12
16   17   10
5    8    11
C:\Java>
```

Program no:3

complex numbers

Aim: Add complex number

Program:

```
public class Complex{
```

```
double a, b;
```

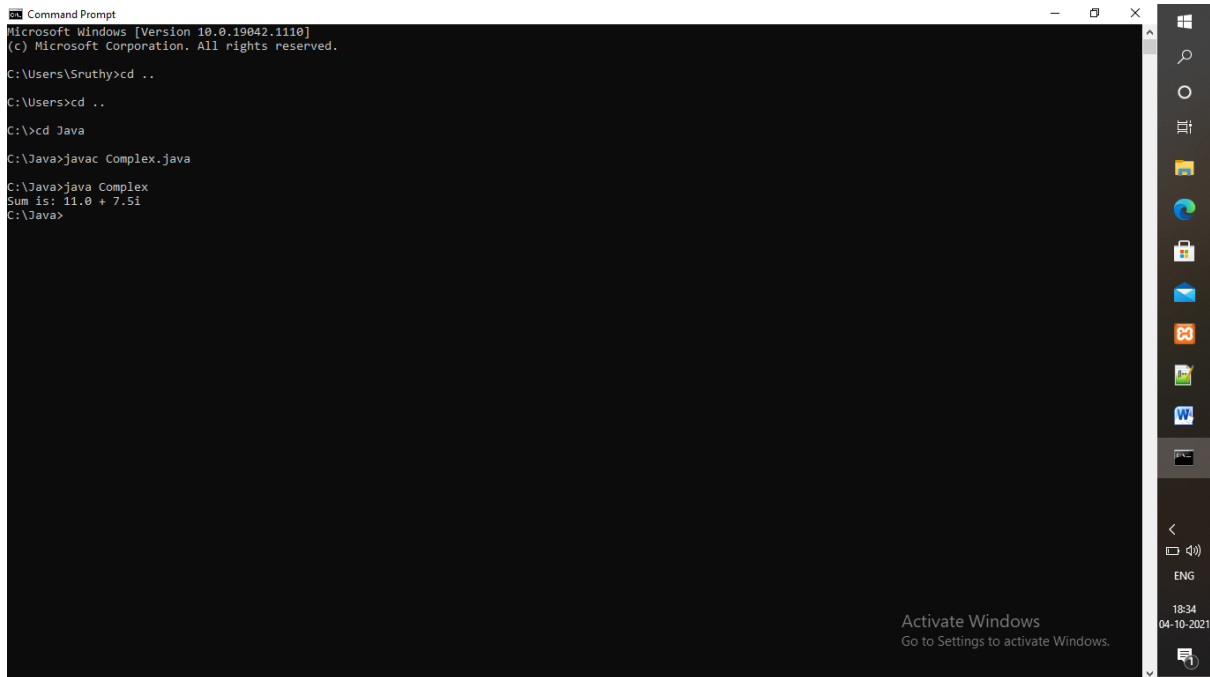
```
Complex(double r, double
i){ this.a = r; this.b = i;
}

public static Complex sum(Complex c1, Complex c2)
{
    Complex temp = new Complex(0, 0);
    temp.a = c1.a + c2.a;
    temp.b = c1.b+ c2.b;
    return temp;
}

public static void main(String args[]) {
    Complex c1 = new Complex(5, 4);
    Complex c2 = new Complex(6, 3.5);
    Complex temp = sum(c1, c2);
    System.out.printf("Sum is: "+ temp.a+" "+ temp.b +"i"); }
}
```

Output:





```
Command Prompt
Microsoft Windows [Version 10.0.19042.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sruthy>cd ..
C:\Users>cd ..
C:\>cd Java
C:\Java>javac Complex.java
C:\Java>java Complex
Sum is: 11.0 + 7.5i
C:\Java>
```

## Program no: 4

### Symmetric

Aim: Read a matrix from the console and check whether it is symmetric or not.

Program:

```
import java.util.Scanner;

public class Symmetric
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
```

```
System.out.println("Enter the no. of rows : ");
```

```
int rows = sc.nextInt();
```

```
System.out.println("Enter the no. of columns : ");
```

```
int cols = sc.nextInt();
```

```
int matrix[][] = new int[rows][cols];
```

```
System.out.println("Enter the elements :");
```

```
for (int i = 0; i < rows; i++)
```

```
{
```

```
for (int j = 0; j < cols; j++)
```

```
{
```

```
matrix[i][j] = sc.nextInt();
```

```
}
```

```
}
```

```
System.out.println("Printing the input matrix :");
```

```
for (int i = 0; i < rows; i++)
```

```
{
```

```
for (int j = 0; j < cols; j++)  
{  
    System.out.print(matrix[i][j]+"\\t");  
}  
System.out.println();  
}
```

```
if(rows != cols)  
{  
    System.out.println("The given matrix is not a square matrix,  
so it can't be symmetric.");  
}
```

```
else
```

```
{  
    boolean symmetric = true;
```

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

```
if(matrix[i][j] != matrix[j][i])
{
    symmetric = false;
    break;
}
}
}
if(symmetric)
{
    System.out.println("The given matrix is symmetric..."); }
else
{
    System.out.println("The given matrix is not symmetric..."); }
}
sc.close();
}
```

Output:

```
Command Prompt
Sum is: 11.0 + 7.5i
C:\Java>javac Symmetric.java
C:\Java>java Symmetric
Enter the no. of rows :
3
Enter the no. of columns :
3
Enter the elements :
5
6
7
1
2
3
7
8
9
Printing the input matrix :
5      6      7
1      2      3
7      8      9
The given matrix is not symmetric...
C:\Java>
```

## Program no:5

cpu

AIM: Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

Program:

```
class CPU { double
```

```
price=27000;
```

```
class
```

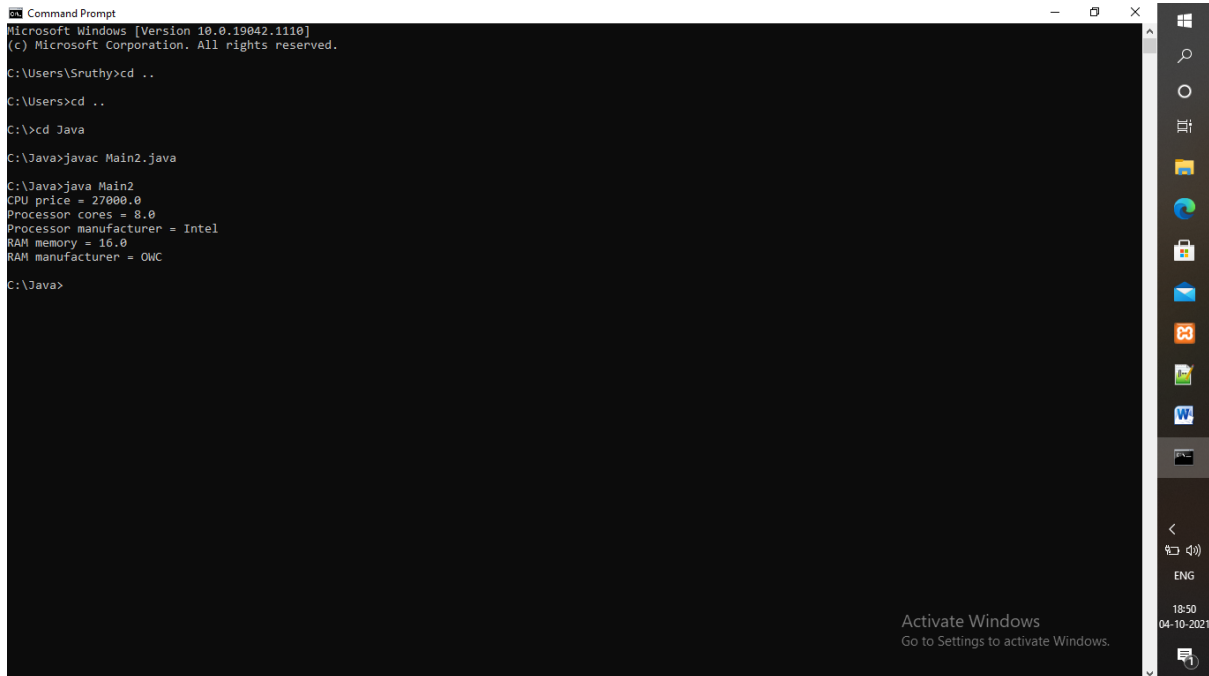
```
Processor{ double
```

```
cores=8;
```

```
String manufacturer="Intel";  
  
}  
  
protected class RAM{  
  
double memory=16;  
  
String manufacturer="OWC";  
  
}  
  
}  
  
public class Main2 {  
  
    public static void main(String[] args) {  
  
        CPU cpu = new CPU();  
  
        CPU.Processor processor = cpu.new Processor();  
  
        CPU.RAM ram = cpu.new RAM();  
  
        System.out.println("CPU price = " + cpu.price);  
  
        System.out.println("Processor cores = " + processor.cores);  
  
        System.out.println("Processor manufacturer = " +  
processor.manufacturer);  
  
        System.out.println("RAM memory = " + ram.memory);  
  
        System.out.println("RAM manufacturer = " +  
ram.manufacturer);  
  
    }  
}
```

}

## Output:



```
Command Prompt
Microsoft Windows [Version 10.0.19042.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sruthy>cd ..
C:\Users>cd ..
C:\>cd Java
C:\Java>javac Main2.java
C:\Java>java Main2
CPU price = 27000.0
Processor cores = 8.0
Processor manufacturer = Intel
RAM memory = 16.0
RAM manufacturer = OWC
C:\Java>
```

## Program no: 6

### Sort String

AIM: Program to Sort strings

Program:

```
public class sortstringss{ public
static void main(String[] args)
{
```

String

```
names[]={"amal","jyothi","college","of","engineering"};
```

```
String temp; int n= names.length;

int i; int j;

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

{

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

{

if(names[i].compareTo(names[j])>0)

{

temp=names[i];

names[i]=names[j];

names[j]=temp; }

}

}

System.out.println("the sorted array of string is :");

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

{

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

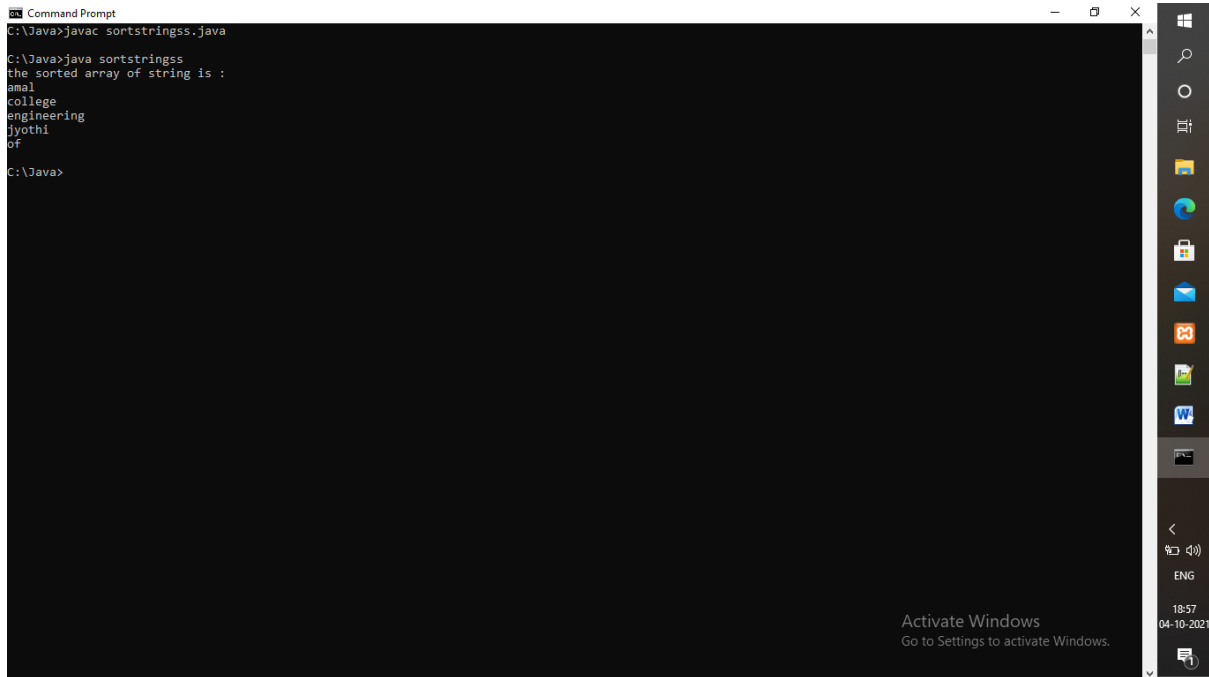
}

}
```



```
}
```

## Output:



```
Command Prompt
C:\Java>javac sortstrings.java

C:\Java>java sortstrings
the sorted array of string is :
amal
college
engineering
jyothi
of
C:\Java>
```

Program no: 7

Search an element

AIM: Search an element in an array.

Program:

```
import java.util.*;

public class search{

public static void main(String[] args)

{

int n,i,b,flag=0;
```

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

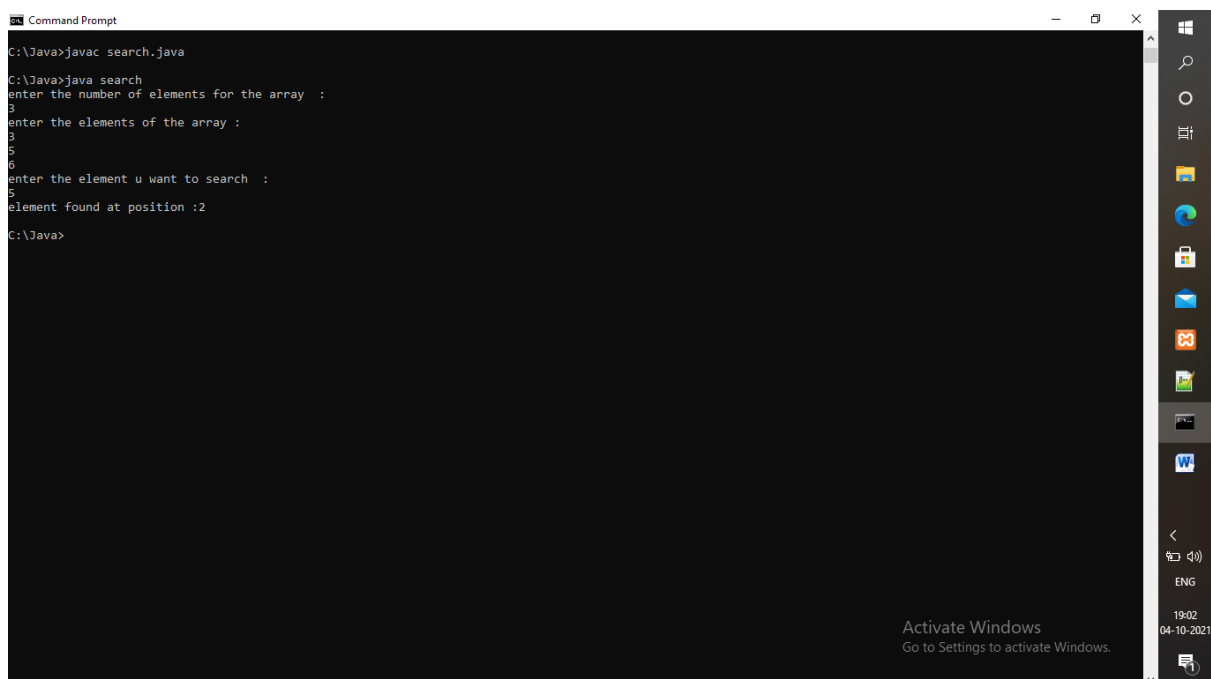
System.out.println("enter the number of elements for the
array
:"); n=s.nextInt(); int a[]=new int[n];

System.out.println("enter the elements of the array :");
for(i=0;i<n;i++)
{
a[i]=s.nextInt();
}

System.out.println("enter the element u want to search :");
b=s.nextInt(); for(i=0;i<n;i++)
{
if(a[i]==b)
{
flag=1;
break;
}
else
{
flag=0;
```

```
}  
  
}  
  
if(flag==1)  
{  
  
System.out.println("element found at position :"+(i+1)); }  
  
else  
  
{  
  
System.out.println("element not found");  
  
}  
  
}  
  
}
```

### Output:



```
Command Prompt  
C:\Java>javac search.java  
C:\Java>java search  
enter the number of elements for the array :  
3  
enter the elements of the array :  
3  
5  
6  
enter the element u want to search :  
5  
element found at position :2  
C:\Java>
```

19:02  
04-10-2021  
ENG  
Activate Windows  
Go to Settings to activate Windows.

## Program no: 8

### String manipulations

AIM: Perform string manipulations

Program:

```
public class Sample_String{ public
static void main(String[] args)
{
    String str_Sample = "littlyStar";

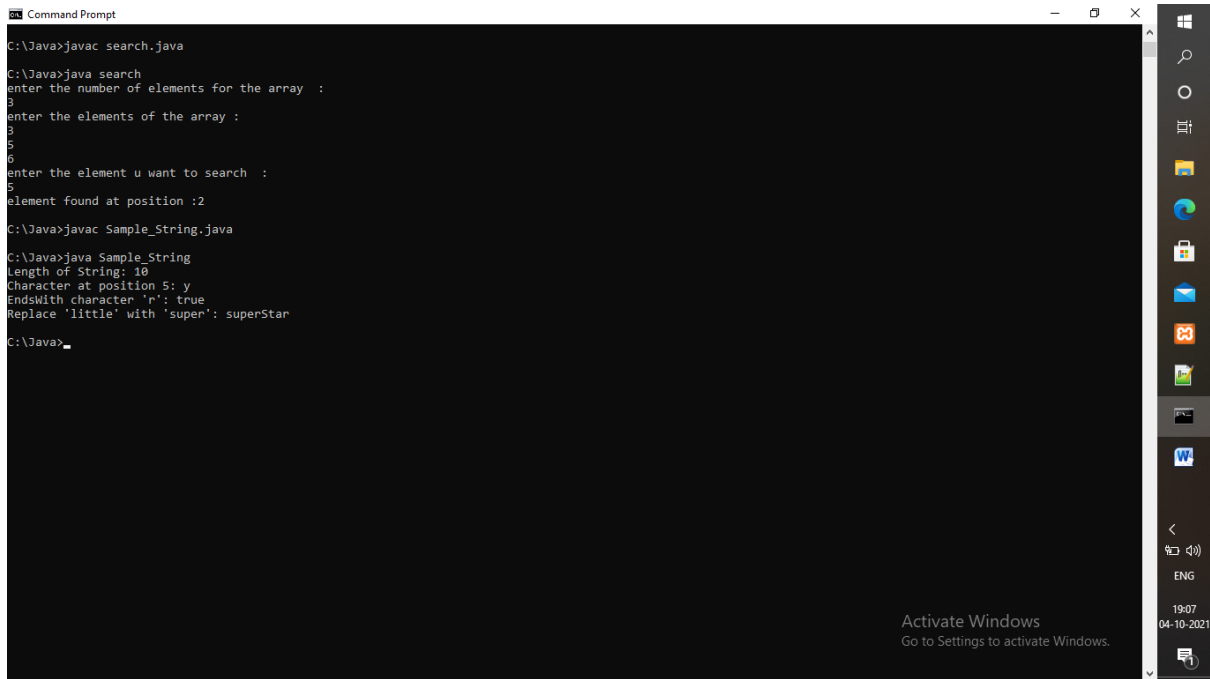
    System.out.println("Length of String: " +
str_Sample.length());

    System.out.println("Character at position 5: " +
str_Sample.charAt(5));

    System.out.println("EndsWith character 'r': " +
str_Sample.endsWith("r"));

    System.out.println("Replace 'little' with 'super': " +
str_Sample.replace("littly", "super"));
}
}
```

Output:



```
C:\Java>javac search.java
C:\Java>java search
enter the number of elements for the array :
5
enter the elements of the array :
3
5
6
enter the element u want to search :
5
element found at position :2
C:\Java>javac Sample_String.java
C:\Java>java Sample_String
Length of String: 10
Character at position 5: y
EndsWith character 'n': true
Replace 'little' with 'super': superStar
C:\Java>
```

## Program no: 9

### Area of shapes

AIM: Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

Program:

```
public class shape
```

```
{
```

```
int s,as,ar;
```

```
public void area(int a)
```

```
{
```

```
s=a;

as=a*a;

System.out.println("area of square is"+as);

}

public void area(double r)

{

double radi=r; double

ac=(22/7)*radi*radi;

System.out.println("area of circle is"+ac);

}

public void area(int l,int w)

{

int len=l;

int wid=w;

ar=len*wid;

System.out.println("area of rectangle"+ar);

}

public void area(int h,double r)

{

int he=h;
```

```
double rad=r;

double acy=(2*(22/7)*rad*he)+((22/7)*rad*rad);

System.out.println("area of cylinder"+acy);

}

public static void main(String[] args) {

shape o=new shape();

o.area(6);//area of square

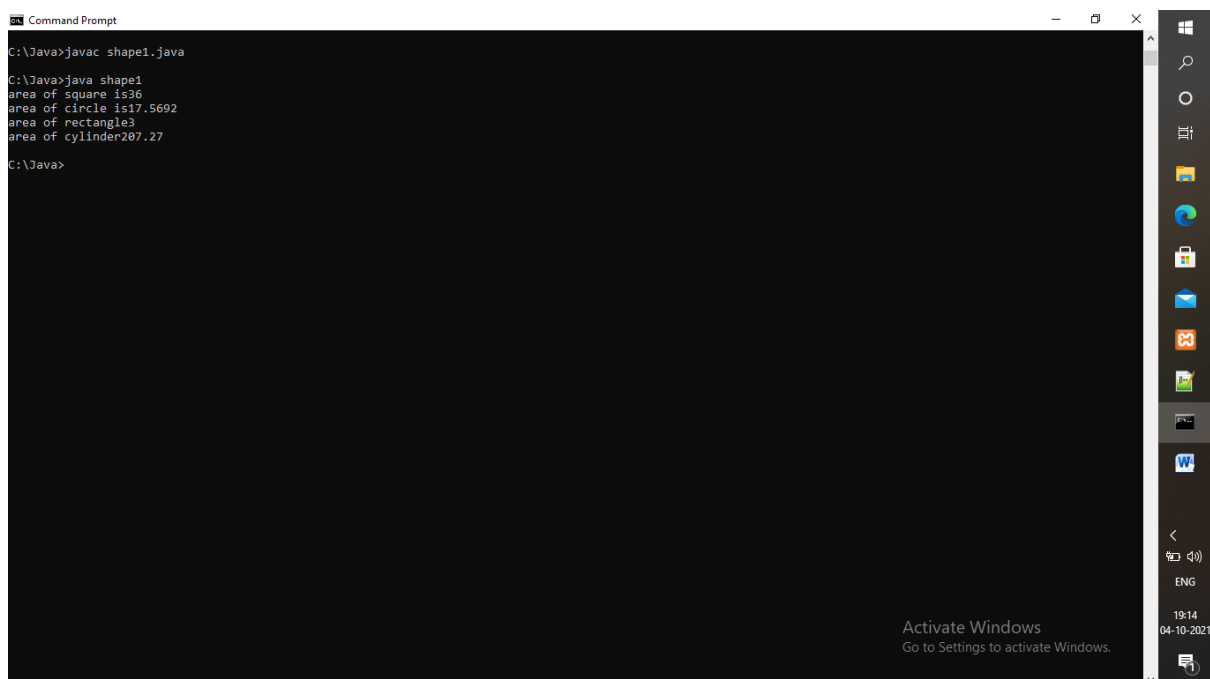
o.area(2.42);//area of circle

o.area(3,1);//area of rectangle

o.area(5,4.7);

} }
```

### Output:



```
Command Prompt
C:\Java>javac shape1.java
C:\Java>java shape1
area of square is36
area of circle is17.5692
area of rectangle3
area of cylinder287.27
C:\Java>
```

Activate Windows  
Go to Settings to activate Windows.

19:14  
04-10-2021

## Program no: 10

### Employee

AIM: Create a class 'Employee' with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class 'Teacher' that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

Program:

```
import java.util.Scanner;

class Person
{
    String name,gender,address;

    int age;

    public Person(String name, String gender, String address, int
age) { super();

    this.name = name; this.gender = gender; this.address =
address; this.age = age;

}

}
```



```
class Employee extends Person {  
    int empid;  
  
    String company_name,qualification;  
  
    double salary;  
  
    public Employee(String name, String gender, String address,  
        int age, int empid, String company_name,  
        String qualification, double salary) {  
        super(name, gender, address, age);  
        this.empid = empid;  
        this.company_name = company_name;  
        this.qualification = qualification;  
        this.salary = salary;  
    }  
}  
  
class Teacher extends Employee  
{  
  
    String subject,department;  
  
    int teacherid;  
  
    public Teacher(String name, String gender, String address, int  
        age, int empid, String company_name,
```

```
String qualification, double salary, String subject, String
department, int teacherid) {

super(name, gender, address, age, empid, company_name,
qualification, salary);

this.subject = subject;

this.department = department;

this.teacherid = teacherid;

}

void display()

{

System.out.println("....Personal details...");

System.out.println(" Name : "+this.name+" Gender :
"+this.gender+" Age : "+this.age);

System.out.println("...Employee details....");

System.out.println("Empid : "+this.empid +" company_name
: "+this.company_name+" Salary : "+this.salary+" Address :
"+this.address+" qualification : "+this.qualification);

System.out.println("...Teacher's details...");

System.out.println(" teacherid : "+this.teacherid+ "
department : "+this.department+" Subjects : "+this.subject);

}
```

```
}  
  
public class Main {  
    public static void main(String[] args) {  
        Scanner s=new Scanner(System.in);  
  
        int n;  
  
        System.out.println("Enter number of Teachers : ");  
        n=s.nextInt();  
  
        Teacher obj[]=new Teacher[n];  
  
        for(int i=0;i<n;i++) {  
  
            System.out.println("Enter the person name:");  
  
            String nam1=s.next();  
  
            System.out.println("Enter the Gender: ");  
  
            String gen1=s.next();  
  
            System.out.println("Enter the Address: ");  
  
            String adr1=s.next();  
  
            System.out.println("Enter the Age:");  
  
            int age1=s.nextInt();  
  
            System.out.println("Enter the Employee id: ");  
  
            int id1=s.nextInt();  
  
            System.out.println("Enter the Company name: ");
```

```

String cname1=s.next();

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

double sal1=s.nextDouble();

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

String qu1=s.next();

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

int tid1=s.nextInt();

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

String dept1=s.next();

System.out.println("Enter the
Subject:"); String sub1=s.next();

obj[i]=new
Teacher(nam1,gen1,adr1,age1,id1,cname1,qu1,sal1,sub1,de
pt1,tid1); }

System.out.println("\n-----
----- ---\n"); for(int

i=0;i<n;i++) {

obj[i].display();

}

}

```

}

## Output:



```
C:\Program Files\Java\jdk-1.8.0_101\bin>java Main.java
C:\Program Files\Java\jdk-1.8.0_101\bin>
Enter number of Teachers :
1
Enter the person name:
rishi
Enter the Gender:
female
Enter the Address:
bangalore
Enter the Age:
18
Enter the Employee Id:
1234
Enter the Company name:
abc
Enter the Salary:
20000
Enter the Qualification:
mcom
Enter the Teacher Id:
0456
Enter the Department:
commerce
Enter the Subject:
finance

.....
...Personal details...
Name : rishi Gender : female Age : 18
...Employee details...
Empid : 1234 company_name : abc Salary : 20000.0 Address : bangalore qualification : mcom
...Teacher's details...
Teacherid : 0456 Department : commerce Subjects : finance
C:\Program Files\Java\jdk-1.8.0_101\bin>
```

## Program no: 11

### Person

AIM: Create a class 'Person' with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class 'Employee' that inherits the properties of class Person and also contains its own data members like Empid, Company\_name, Qualification, Salary and its own constructor. Create another class 'Teacher' that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

Program:

```
import java.util.*;
```

```
class Employee
```

```
{
```

```
int empid;
```

```
String name,address;
```

```
double salary;
```

```
public Employee(int empid, String name, String address,  
double salary) { this.empid = empid; this.name = name;  
this.address = address; this.salary = salary;
```

```
}
```

```
}
```

```
public class Teacher extends Employee
```

```
{
```

```
String subject,department;
```

```
public Teacher(int empid, String name, String address, double  
salary,String department,String subject ) { super(empid,  
name, address, salary); this.subject = subject;  
this.department = department;
```

```
}
```

```
void display()
```

```
{  
    System.out.println("Empid : "+this.empid+" Name :  
    "+this.name+" Salary : "+this.salary+" Address :  
    "+this.address+" department : "+this.department+" Subjects  
    : "+this.subject);  
}  
  
public static void main(String[] args) {  
    Scanner sc=new Scanner(System.in);  
  
    int n;  
  
    System.out.println("Enter number of Teachers : ");  
  
    n=sc.nextInt();  
  
    Teacher obj[]=new  
    Teacher[n]; for(int i=0;i<n;i++)  
    { int j = i+1;  
  
        System.out.print("Enter Empid of teacher "+j+" : ");  
  
        int Empid = sc.nextInt();  
  
        System.out.print("Enter Name of teacher "+j+" : ");  
  
        String Name = sc.next();  
  
        System.out.print("Enter Salary of teacher "+j+" : ");  
  
        double Salary = sc.nextDouble();  
  
        System.out.print("Enter Address of teacher "+j+" : ");
```

```

String Address = sc.next();

System.out.print("Enter department of teacher "+j+" : ");

String department =sc.next();

System.out.print("Enter Subjects of teacher "+j+" : ");

String Subjects =sc.next();

obj[i] = new Teacher(Empid, Name, Address, Salary,
department, Subjects);

}

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

System.out.println("Teacher's List \n");

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

obj[i].display();

}

}

}

```

Output:



```
Command Prompt
C:\Java>javac Teacher.java

C:\Java>java Teacher
Enter number of Teachers :
3
Enter Empid of teacher 1 : 1
Enter Name of teacher 1 : jincy
Enter Salary of teacher 1 : 50000
Enter Address of teacher 1 : kottayam
Enter department of teacher 1 : mca
Enter Subjects of teacher 1 : software engineering
Enter Empid of teacher 2 : Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:939)
    at java.base/java.util.Scanner.next(Scanner.java:1594)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
    at Teacher.main(Teacher.java:28)

C:\Java>
```

## Program no: 12

### BOOKS

AIM: Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.

Program:

```
import java.util.Scanner;
```

```
class Publisher {
```

```
String Pubname;
```

```
Publisher()
```

```
{
```

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

```
System.out.println("Enter publisher name");
Pubname=s.next();
}
}

class Book extends Publisher
{
String title, author;
int price;
Book()
{
Scanner s=new Scanner(System.in);
System.out.println("Enter Title of the book"); title=s.next();
System.out.println("Enter Author's name"); author=s.next();
System.out.println("Enter price");
price=s.nextInt();
}
}

class Literature extends Book
{
Literature()
```

```
{  
    System.out.println("Literature Books"); }  
  
void display()  
{  
    System.out.println("Publisher name: "+Pubname);  
    System.out.println("Title of the book: "+title);  
    System.out.println("Author's name: "+author);  
    System.out.println("Price: "+price);  
}  
}  
  
class Fiction extends Literature  
{  
    Fiction()  
    {  
        System.out.println("Friction Books");  
    }  
    void display()  
    {  
        super.display();  
    }  
}
```

```
public static void main(String args[])
{
    int n;

    Scanner s=new Scanner(System.in);

    System.out.println("Enter the No of literature book: ");

    int a=s.nextInt();

    Literature L[]=new Literature[a];

    for(int i=0;i<a;i++)
    {
        L[i]=new Literature();
    }

    System.out.println("Enter the No of Fiction book: ");

    int b=s.nextInt();

    Fiction F[]=new Fiction[b];

    for(int i=0;i<b;i++) {
        F[i]=new Fiction(); }

    int no;

    System.out.println("Enter your choice of book");
    no=s.nextInt();

    int type =no;

    switch (no) {
```

case 1:

```
System.out.println(".....Details of literature books");
```

```
for(int i=0;i<a;i++) L[i].display();
```

```
break;
```

case 2:

```
System.out.println(".....Details of fiction books");
```

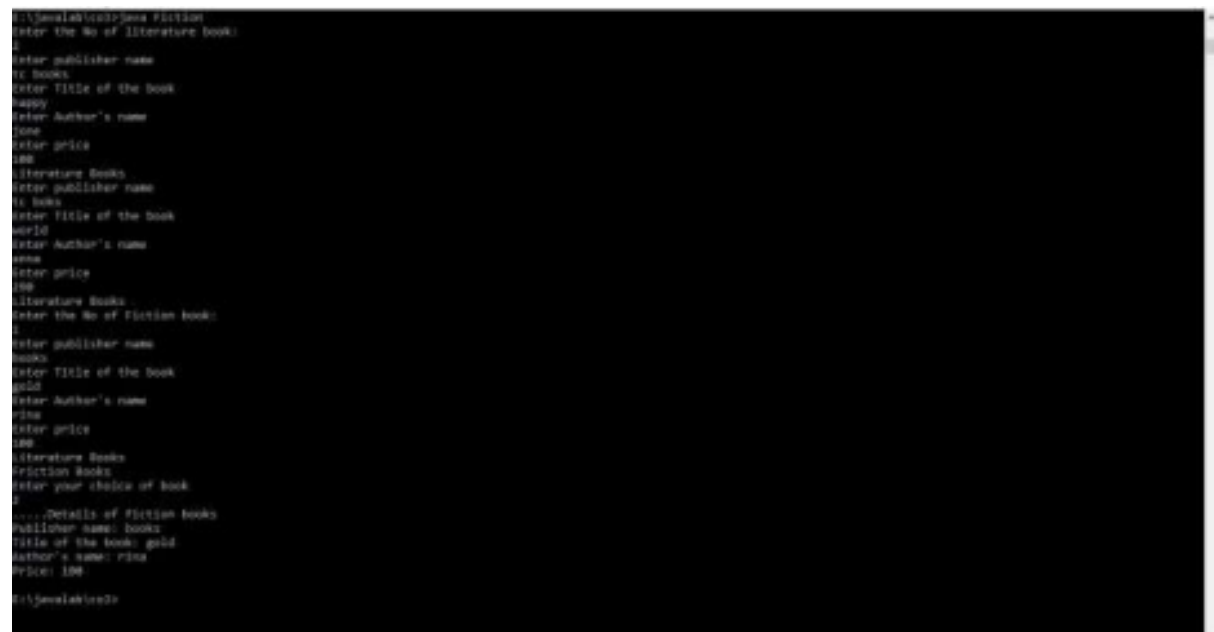
```
for(int i=0;i<b;i++) F[i].display();
```

```
break;
```

```
default: System.out.println("Wrong input");
```

```
} } }
```

Output:



```
C:\jswalake>java Fiction
Enter the No of literature book:
1
Enter publisher name
1: books
Enter Title of the book
happy
Enter Author's name
jane
Enter price
100
Literature Books
Enter publisher name
1: books
Enter Title of the book
world
Enter Author's name
anna
Enter price
100
Literature Books
Enter the No of Fiction book:
1
Enter publisher name
books
Enter Title of the book
gold
Enter Author's name
rina
Enter price
100
Literature Books
Fiction Books
Enter your choice of book
2
.....Details of fiction books
Publisher name: books
Title of the book: gold
Author's name: rina
Price: 100
C:\jswalake>
```

### Program no: 13

#### Student and sports

AIM: Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

Program:

```
interface student
```

```
{
```

```
    void stresult();
```

```
}
```

```
interface sports
```

```
{
```

```
    void spresult();
```

```
}
```

```
class result implements student,sports{
```

```
    public void spresult()
```

```
{
```

```
        String hundred="First";
```

```
        String twohundred="Second";
```

```
        String fivehundred="First";
```

```
String relay="Second";

System.out.println("Sports Result");

System.out.println("Hundred Meter:"+hundred);

System.out.println("Two Hundred Meter:"+twohundred);
System.out.println("Five Hundred Meter:"+fivehundred);
System.out.println("Relay:"+relay);

}

public void stresultt()

{

int physics=30;

int

chemistry=40;

int maths=45; int

english=50; int

computer=50;

System.out.println("Marks");

System.out.println("Physics:"+physics);

System.out.println("Chemistry:"+chemistry);

System.out.println("Mathematics:"+maths);

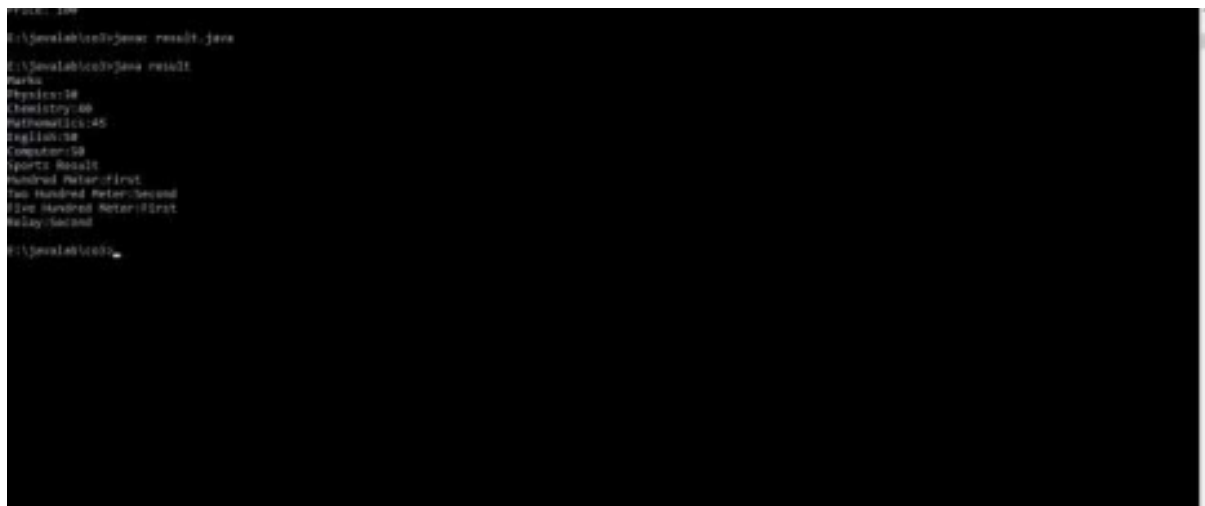
System.out.println("English:"+english);
```

```
        System.out.println("Computer:"+computer);
    }

    public static void main(String[] args)
    {
        result r = new result();

        r.stresultt();
        r.spresult();
    }
}
```

### Output:

A screenshot of a Java program's output in a terminal window. The output shows the marks for a student named Peter across various subjects: Maths (58), Chemistry (48), Mathematics (45), English (48), and Computer (58). It also displays sports results for Peter: 100m (First), 200m (Second), 400m (First), and Relay (Second). The terminal window has a black background with white text.

```
C:\javabasics>java result.java
C:\javabasics>java result
marks
Maths:58
Chemistry:48
Mathematics:45
English:48
Computer:58
Sports Result
100m Peter:First
200m Peter:Second
400m Peter:First
Relay:Second
C:\javabasics>
```

### Program no:14

Area And Perimeter



AIM: Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

PROGRAM:

```
public class shape
{
    int s,as,ar;

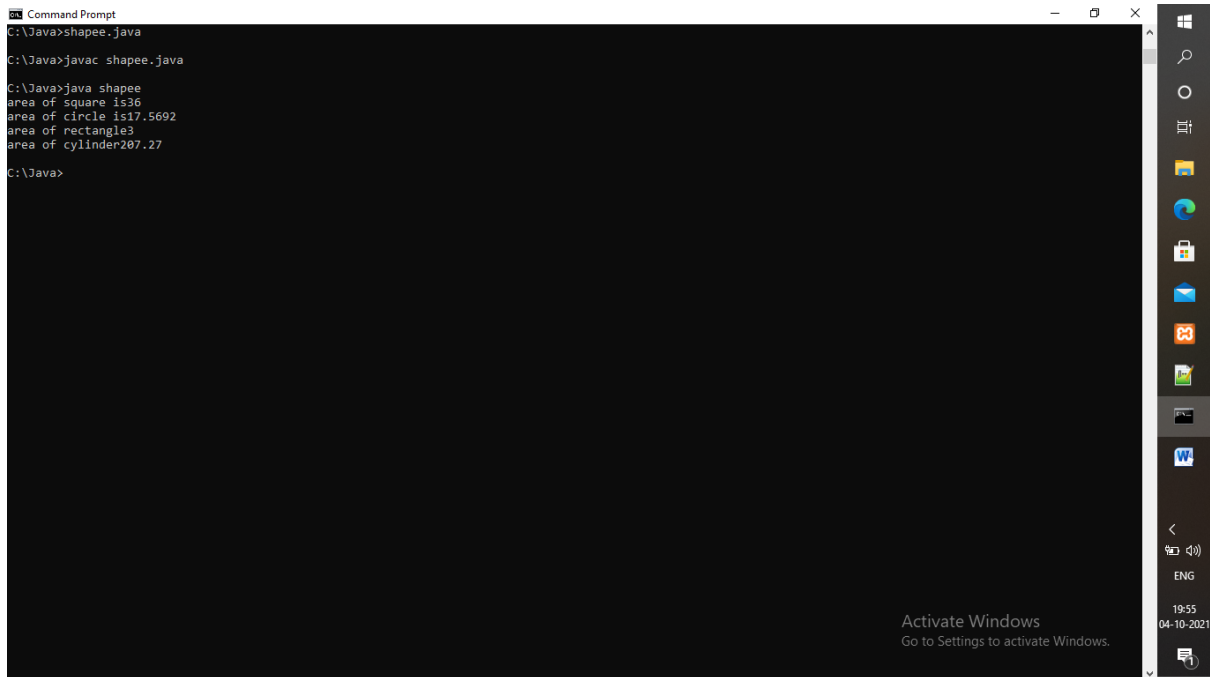
    public void area(int a)
    {
        s=a;
        as=a*a;
        System.out.println("area of square is"+as);
    }

    public void area(double r)//area of circle
    {
        double radi=r; double
        ac=(22/7)*radi*radi;
        System.out.println("area of circle is"+ac);
    }

    public void area(int l,int w)//area of rectangle
```

```
{  
int len=l;  
int wid=w;  
ar=len*wid;  
System.out.println("area of rectangle"+ar);  
}  
  
public void area(int h,double r)  
{  
int he=h; double rad=r; double  
acy=(2*(22/7)*rad*he)+((22/7)*rad*rad);  
System.out.println("area of cylinder"+acy);  
}  
  
public static void main(String[] args)  
{  
shape o=new shape();  
o.area(6);  
o.area(2.42);  
o.area(3,1);  
o.area(5,4.7);  
}
```

```
}
```



```
Command Prompt
C:\Java>shapee.java
C:\Java>javac shapee.java
C:\Java>java shapee
area of square is26
area of circle is17.5692
area of rectangle3
area of cylinder207.27
C:\Java>
```

The screenshot shows a Windows desktop with a taskbar on the right. The taskbar contains icons for File Explorer, Edge, Mail, and other applications. The Command Prompt window is open, displaying the execution of a Java program named 'shapee.java'. The program calculates the area of a square (26), a circle (17.5692), a rectangle (3), and a cylinder (207.27). The Windows logo is visible in the top right corner of the taskbar. At the bottom right of the Command Prompt window, there is a watermark that says 'Activate Windows Go to Settings to activate Windows.'

## Program no: 15

### ProductBill

AIM: Prepare bill with the given format using calculate method from interface. Order No. Date : Product Id Name Quantity unit price Total 101 A 2 25 50 102 B 1 100 100 Net. Amount 150

Program:

```
interface bill
```

```
{
```

```
int productdetails();
```

```
}
```

```
class product1 implements bill{ int id =  
101,quantity= 2,unit=25,total=0;  
String name="A";  
public int productdetails()  
{  
total = quantity * unit;  
System.out.println("Product Id :"+id);  
System.out.println("Name :"+name);  
System.out.println("Quantity :"+quantity);  
System.out.println("Unit price :"+unit);  
System.out.println("Total :"+total); return(total);  
}  
}  
  
class product2 implements bill{ int id = 102,quantity=  
1,unit=100,total=0; String name="B";  
public int productdetails()  
{  
total = quantity * unit;  
System.out.println("Product Id :"+id);  
System.out.println("Name :"+name);  
System.out.println("Quantity :"+quantity);
```

```
System.out.println("Unit price :"+unit);
System.out.println("Total :"+total); return(total);

}

}

public class productbill

{

public static void main(String[] args)

{

product1 p1 = new product1();

product2 p2 = new product2();

int t1= p1.productdetails();

int t2= p2.productdetails();

int t3=t1+t2;

System.out.println("Net. Amount :"+t3);

} }
```

Output:

```
Command Prompt
C:\Java>javac productbill.java
C:\Java>java productbill
Product Id :101
Name :A
Quantity :2
Unit price :25
Total :50
Product Id :102
Name :B
Quantity :1
Unit price :100
Total :100
Net. Amount :150
C:\Java>_
```

Activate Windows  
Go to Settings to activate Windows.

20:01  
04-10-2021

Program no: 17

Average

AIM: Find the average of N positive integers, raising a user defined exception for each negative input

Program:

```
import java.util.Scanner;

import java.util.InputMismatchException;

public class TestDemo

{

public static void main(String args[])

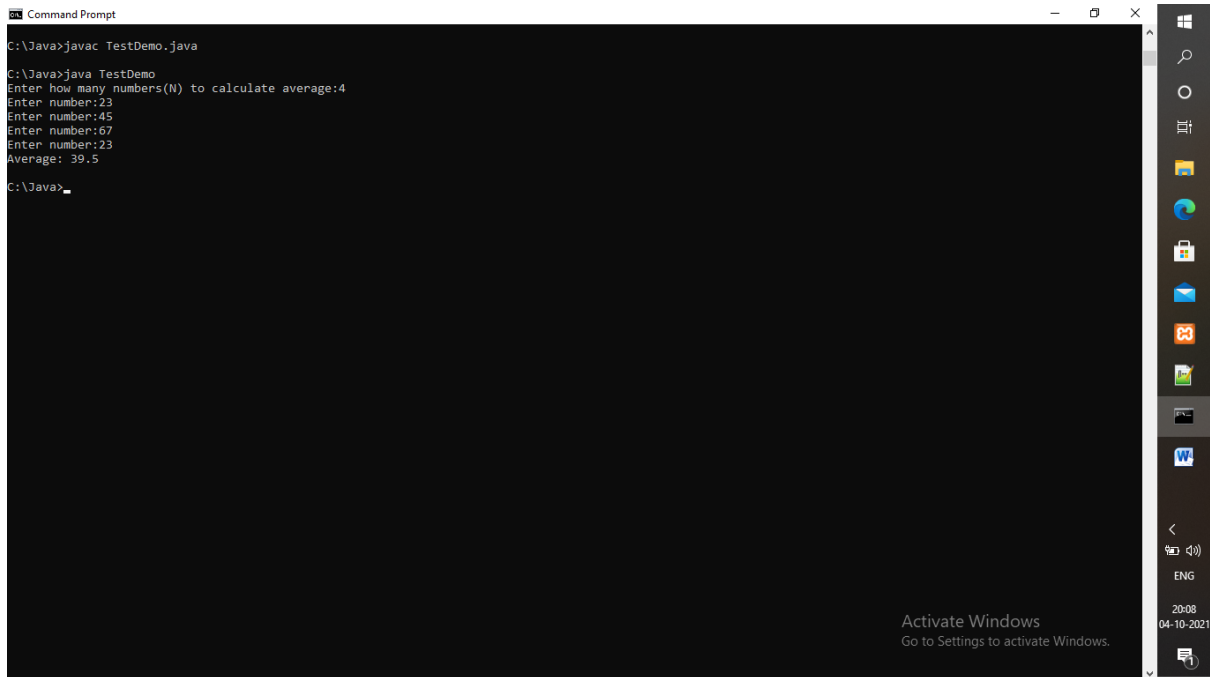
{
```

```
double total = 0, N, userInput; Scanner
input = new Scanner(System.in);
while (true)
{
System.out.print("Enter how many numbers(N) to calculate
average:");
userInput = input.nextDouble();
if (userInput > 0) {
N = userInput;
break;
}
else
System.out.println("N must be positive.");
}
for (int i = 0; i < N; i++)
{
while (true)
{
System.out.print("Enter number:");
try
```

```
{  
    userInput = input.nextDouble();  
    total += userInput;  
    break;  
}  
catch (InputMismatchException e)  
{input.nextLine();  
    System.out.println("Input must be a number. Try again");  
}  
}  
}  
System.out.println("Average: "+ total / N);  
}  
}
```

Output:





```
Command Prompt
C:\Java>javac TestDemo.java
C:\Java>java TestDemo
Enter how many numbers(N) to calculate average:4
Enter number:23
Enter number:45
Enter number:67
Enter number:23
Average: 39.5
C:\Java>
```

Program no: 18

Thread

AIM: Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers.

Implement using threads. (Thread class)

Program:

```
import java.util.Scanner; class
```

```
MulTable extends Thread{
```

```
public void run() { int num = 5;
```

```
System.out.printf("_____Multiplication Table of 5_____\\n");
```

```
for(int i = 1; i <= 10; ++i)
```

```
{
```

```
System.out.printf("%d * %d = %d \\n", num, i, num * i);
```

```
}
```

```
}
```

```
}
```

```
class PrimeNo extends Thread{
```

```
public void run() {
```

```
int i, j, flag;
```

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

```
System.out.println("\n_____To generate first N prime  
numbers_____");
```

```
System.out.println("Enter the limit (N):");
```

```
int N = s.nextInt();
```

```
System.out.println("Prime numbers between 1 and " + N + "  
are:");
```

```
for (i = 1; i <= N; i++)
```

```
{
```

```
if (i == 1 || i == 0)
```

```
continue;
```

```
flag = 1;
```

```
for (j = 2; j <= i / 2; ++j)
```

```
{
```

```
if (i % j == 0)
{
flag = 0;
break;
}
}

if (flag == 1)
System.out.print(i + " ");
}
}
}
```

```
public class ThreadClass { public static void main(String[]
args) throws InterruptedException { MulTable m = new
MulTable();

m.start();

m.sleep(200);

PrimeNo p = new PrimeNo();

p.start();

p.sleep(200);

}
```

```
}
```

### Output:

Program no: 19

Fibonacci

AIM: Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range.

Implement using threads. (Runnable Interface)

Program:

```
public class Mythread {  
    public static void main(String[] args) {  
        Runnable r = new Runnable1();  
        Thread t = new Thread(r);  
        t.start();  
        Runnable r2 = new Runnable2();  
        Thread t2 = new Thread(r2);  
        t2.start();  
    }  
}  
  
class Runnable2 implements Runnable{  
    public void run(){
```

```
for(int i=0;i<11;i++){  
    if(i%2 == 1)  
        System.out.println(i);  
    }  
}  
  
class Runnable1 implements Runnable{  
    public void run(){ int  
n1=0,n2=1,n3,i,count=10;  
System.out.print(n1+" "+n2);//printing 0 and 1  
for(i=2;i<count;++i  
{  
    n3=n1+n2;  
    System.out.print(" "+n3);  
n1=n2;  
n2=n3;  
}  
}  
}
```

Output:

```
Command Prompt
Product Id :101
Name :A
Quantity :2
Unit price :25
Total :50
Product Id :102
Name :B
Quantity :1
Unit price :100
Total :100
Net. Amount :150

C:\Java>javac TestDemo.java

C:\Java>java TestDemo
Enter how many numbers(N) to calculate average:4
Enter number:23
Enter number:45
Enter number:67
Enter number:23
Average: 39.5

C:\Java>javac Thread.java
Thread.java:14: error: unclosed string literal
System.out.println("\n_____To generate first N prime
                  ^
Thread.java:15: error: unclosed string literal
numbers_____");
                ^
Thread.java:15: error: not a statement
numbers_____");
                ^
3 errors

C:\Java>javac MyThread.java
MyThread.java:25: error: unclosed string literal
System.out.print("
                ^
MyThread.java:26: error: unclosed string literal
"+n3); n1=n2;
                ^
2 errors

C:\Java>javac MyThread.java
.\Thread.java:14: error: unclosed string literal
System.out.println("\n_____To generate first N prime
                  ^
```

## Program no: 20

### BubbleSort

AIM: Using generic method perform Bubble sort.

Program:

```
public class BubbleSort {

    static void bubbleSort(int[] arr) {

        int n = arr.length;

        int temp = 0;

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

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

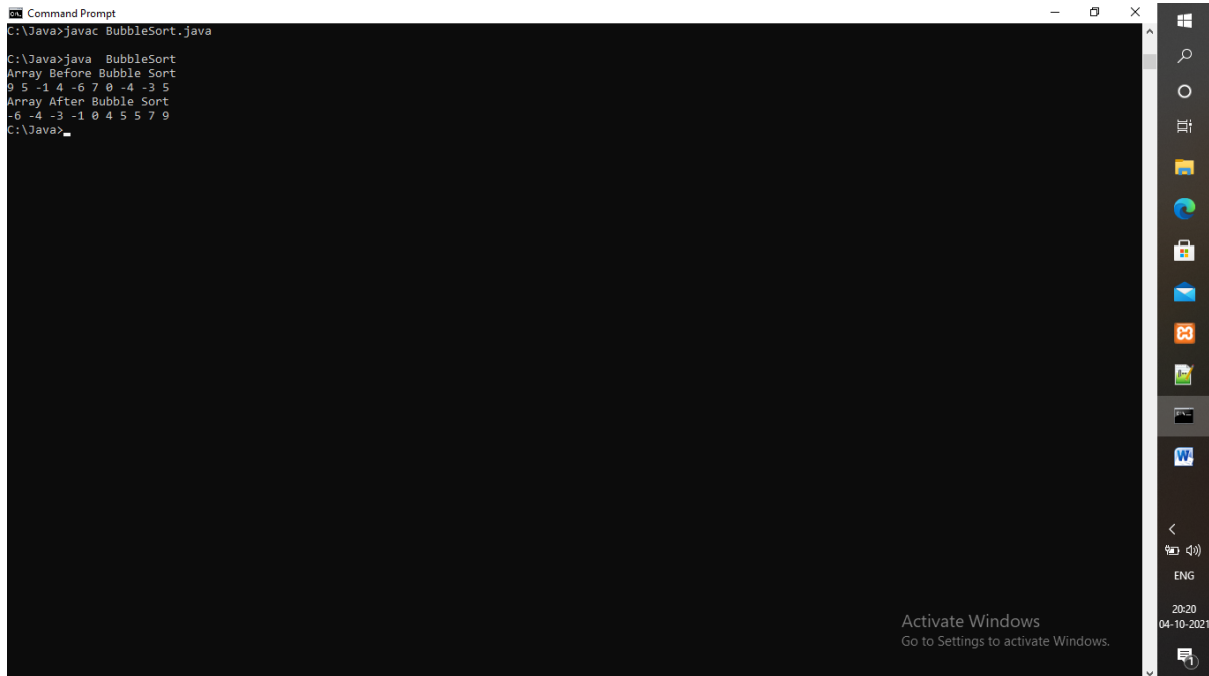
                if(arr[j-1] > arr[j]) {
```

```
        temp = arr[j-1];
        arr[j-1] = arr[j];
        arr[j] = temp;
    }
}
}
}

public static void main(String[] args) {
    int arr[] = { 9, 5, -1, 4, -6, 7, 0, -4, -3, 5 };
    System.out.println("Array Before Bubble Sort");
    for(int i = 0; i < arr.length; i++) {
        System.out.print(arr[i] + " ");
    }
    System.out.println();
    bubbleSort(arr);
    System.out.println("Array After Bubble Sort");
    for(int i = 0; i < arr.length; i++) {
        System.out.print(arr[i] + " ");
    }
}
```

}

## Output:



```
Command Prompt
C:\Java>javac BubbleSort.java

C:\Java>java BubbleSort
Array Before Bubble Sort
0 5 -1 4 -6 7 0 -4 -3 5
Array After Bubble Sort
-6 -4 -3 -1 0 4 5 5 7 9
C:\Java>
```

## Program no: 21

### Array List

AIM: Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

Program:

```
import java.util.*;
```

```
public class ArrayList1 {
```

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

```
        ArrayList<String> obj = new ArrayList<String>();
```



```
obj.add("Aju");  
obj.add("Hanna");  
obj.add("Chandhu");  
obj.add("Sonu");  
obj.add("Anu");  
System.out.println("Original ArrayList:");  
for(String str:obj)  
    System.out.println(str);  
obj.add(0, "Rajeev");  
obj.add(1, "Appu");  
    System.out.println("ArrayList after add operation:");  
for(String str:obj)  
    System.out.println(str);  
obj.remove("Chandhu");  
obj.remove("Hanna");  
System.out.println("ArrayList after remove operation:");  
for(String str:obj)  
    System.out.println(str);  
obj.remove(1);  
    System.out.println("Final ArrayList:");
```

```

        for(String str:obj)

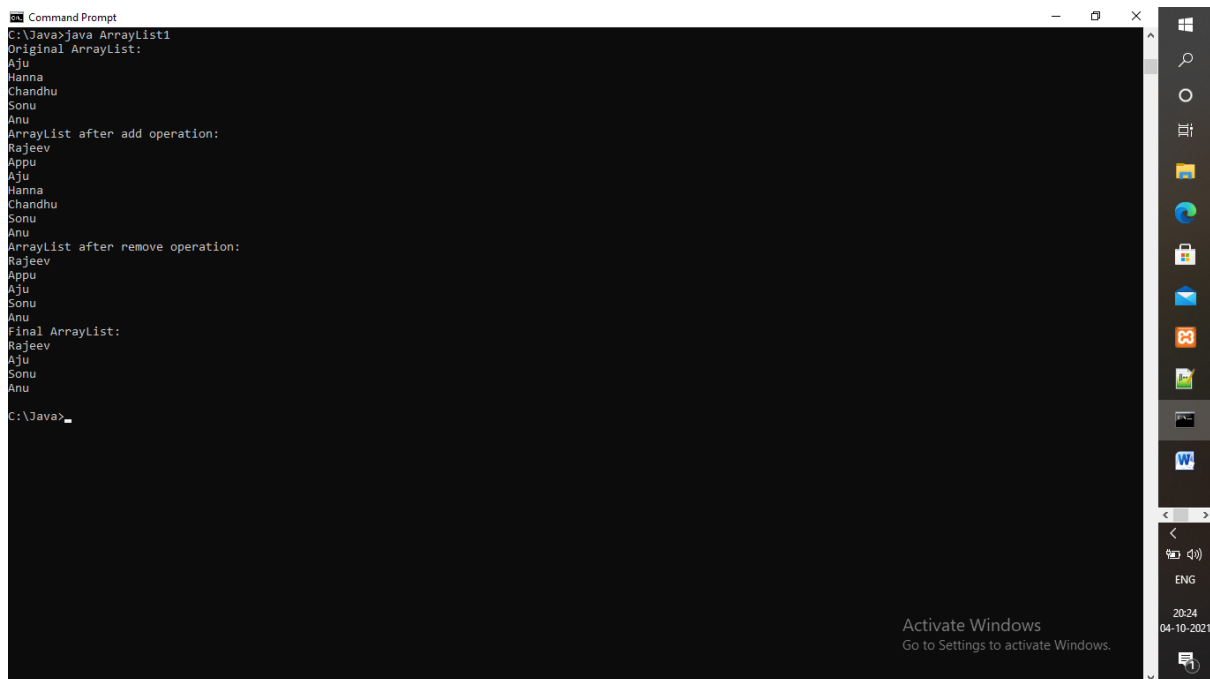
            System.out.println(str);

    }

}

```

### Output:



```

C:\Java>java ArrayList1
Original ArrayList:
Aju
Hanna
Chandhu
Sonu
Anu
ArrayList after add operation:
Rajeev
Apu
Aju
Hanna
Chandhu
Sonu
Anu
ArrayList after remove operation:
Rajeev
Apu
Aju
Sonu
Anu
Final ArrayList:
Rajeev
Aju
Sonu
Anu
C:\Java>

```

### Program no: 22

#### Linked List

AIM: Program to remove all the elements from a linked list

Program:

```

import java.util.*;

public class removelink {

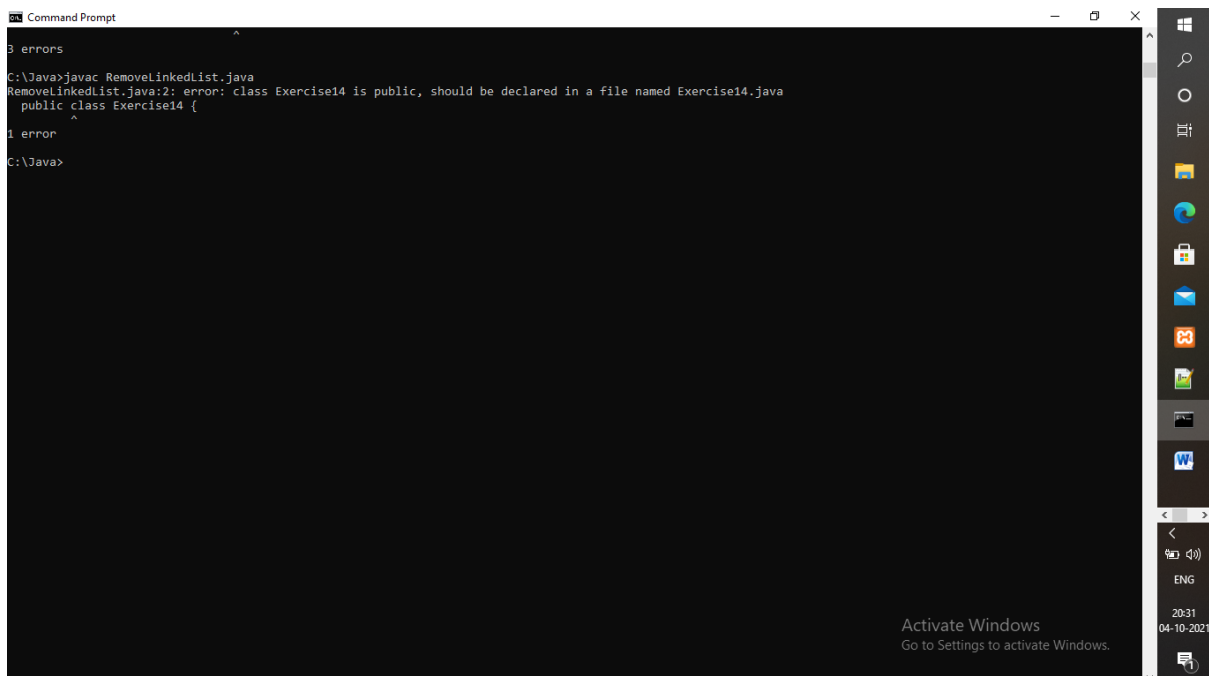
public static void main(String[] args) {

LinkedList<String> l_list = new

```

```
LinkedList<String>();  
values in the linked list l_list.add("hello");  
l_list.add("how"); l_list.add("are");  
l_list.add("you"); l_list.add("?");  
System.out.println("The Original linked list: " + l_list);  
l_list.clear();  
System.out.println("The New linked list: " + l_list);  
}  
}
```

### Output:



The screenshot shows a Windows Command Prompt window with the following text:

```
3 errors  
C:\Java>javac RemoveLinkedList.java  
RemoveLinkedList.java:2: error: class Exercise14 is public, should be declared in a file named Exercise14.java  
    public class Exercise14 {  
        ^  
1 error  
C:\Java>
```

The error message indicates that the class `Exercise14` is public but is not in a file named `Exercise14.java`. The Windows taskbar is visible on the right side of the screen, and an "Activate Windows" watermark is present in the bottom right corner.

Program no: 23

Deque

AIM: Program to demonstrate the addition and deletion of elements in deque

Program:

```
dequeue import java.util.*;

public class DequeExample { public static void main(String[]
args)
{
Deque<String> deque = new LinkedList<String>();
deque.add("Element 1 (Tail)");
deque.addFirst("Element 2 (Head)");
deque.addLast("Element 3 (Tail)");
deque.push("Element 4 (Head)");
deque.offer("Element 5 (Tail)");
deque.offerFirst("Element 6 (Head)");
System.out.println(deque + "\n");
deque.removeFirst();
deque.removeLast();

System.out.println("Deque after removing " +
"first and last: " + deque);
}
}
```

## Output:



```
Command Prompt
C:\Java>javac DequeExample.java
C:\Java>java DequeExample
[Element 6 (Head), Element 4 (Head), Element 2 (Tail), Element 1 (Tail), Element 3 (Tail), Element 5 (Tail)]
Deque after removing first and last: [Element 4 (Head), Element 2 (Head), Element 1 (Tail), Element 3 (Tail)]
C:\Java>
```

## Program no: 24 Map Interface

AIM: . Program to demonstrate the working of Map interface by adding, changing and removing elements.

Program:

```
import java.util.*;

public class MapExample1 {

    public static void main(String[] args) {

        Map map=new HashMap();

        //Adding elements to map

        map.put(1,"Amit");
```

```
map.put(5,"Rahul");  
map.put(2,"Jai");  
map.put(6,"Amit");  
  
//Traversing Map  
  
Set set=map.entrySet();//Converting to Set so that we can  
traverse  
  
Iterator itr=set.iterator();  
while(itr.hasNext()){  
    //Converting to Map.Entry so that we can get key and  
value separately  
    Map.Entry entry=(Map.Entry)itr.next();  
    System.out.println(entry.getKey()+" "+entry.getValue());  
}  
}  
}
```

Output:

```
1 Amit  
2 Jai  
5 Rahul  
6 Amit
```

Program no: 25

## Map Interface

AIM: Program to Convert HashMap to TreeMap Program:

```
import java.util.*;

import java.util.stream.*;

public class HT {

    public static void main(String args[]) {

        Map<String, String> map = new

        HashMap<>();

        map.put("1", "One");

        map.put("2", "Two");

        map.put("3", "Three");

        map.put("4", "Four");

        map.put("5", "Five");

        map.put("6", "Six");

        map.put("7", "Seven");

        map.put("8", "Eight");

        map.put("9", "Nine");

        System.out.println("HashMap = " + map);

        Map<String, String> treeMap = new TreeMap<>();

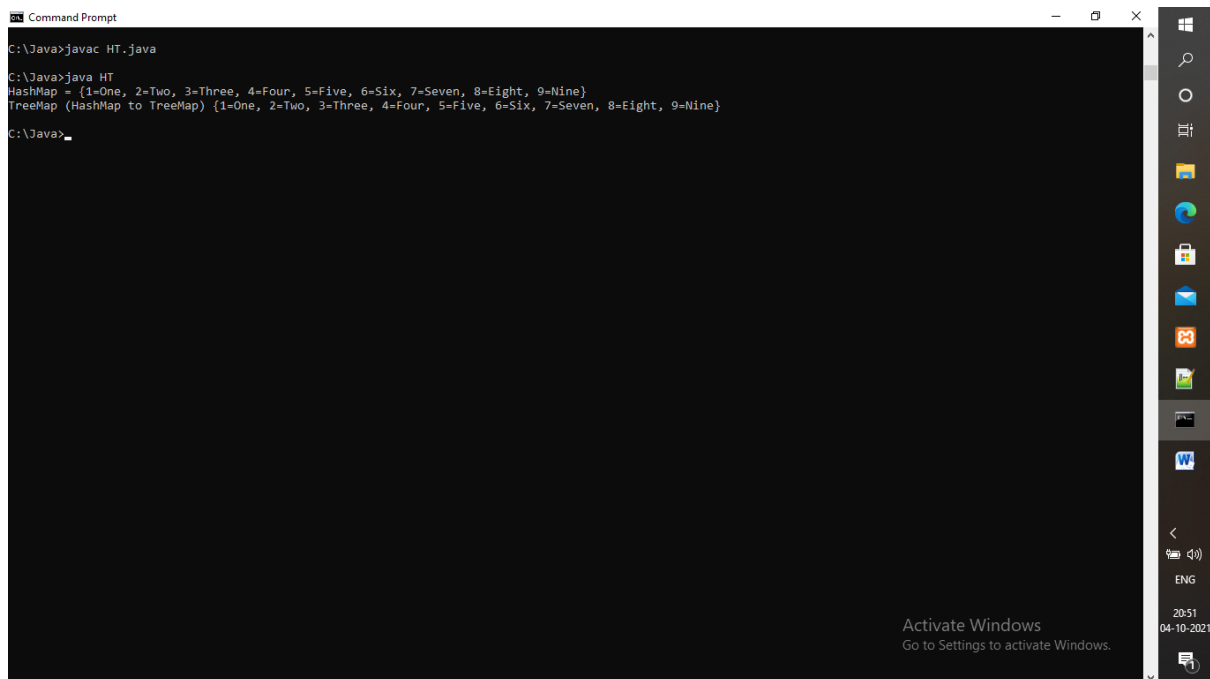
        treeMap.putAll(map);
```

```
System.out.println("TreeMap (HashMap to TreeMap) " +  
treeMap);
```

```
}
```

```
}
```

Output:



```
Command Prompt  
C:\Java>javac HT.java  
  
C:\Java>java HT  
HashMap = {1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight, 9=Nine}  
TreeMap (HashMap to TreeMap) {1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight, 9=Nine}  
  
C:\Java>
```

## Program no: 26

Stack operations

AIM: Program to create a generic stack and do the Push and Pop operations.

Program:

```
java.io.*;
```

```
java.util.*;
```

```
stack<T> {
```



```
ArrayList<T> A;

int top = -1;

int size;

stack(int size)

{

this.size = size;

// Creating array of Size = size this.A = new
ArrayList<T>(size);

}

{

if (top + 1 == size)

{

System.out.println("Stack Overflow");

}

else

{

if (A.size() > top) A.set(top, X);

else

A.add(X);

}
```

```
}  
  
T top()  
  
{  
  
    // If stack is empty if  
    (top == -1)  
  
    {  
  
        // Display message when there are no elements in // the  
        stack System.out.println("Stack Underflow");  
  
        return null;  
  
    }  
  
    // else elements are present so  
    // return the topmost element  
    else return A.get(top);  
  
    }  
  
    // Method 3  
  
    // To delete last element of stack void  
    pop()  
  
    {  
  
        // If stack is empty if  
        (top == -1)
```

```
{  
    System.out.println("Stack Underflow");  
}  
  
else  
    top--;  
}  
  
empty() { return top == -1;  
}  
  
String toString()  
{  
    String Ans = ""; for (int  
        i = 0; i < top; i++)  
    {  
        Ans += String.valueOf(A.get(i)) + "->";  
    }  
    Ans += String.valueOf(A.get(top)); return  
    Ans;  
}  
}  
  
// Main Class public class GFG { // main
```

```
driver method public static void
main(String[] args)
{
    Declaring objects of Integer type
    stack<Integer> s1 = new stack<>(3);
    // Pushing elements to integer stack - s1
    // Element 1 - 10
    s1.push(10); //
    Element 2 - 20
    s1.push(20); //
    Element 3 - 30
    s1.push(30);
    // Print the stack elements after pushing the // elements
    System.out.println("s1 after pushing 10, 20 and 30 :\n" + s1);
    // Now, pop from
    stack s1 s1.pop();
    // Print the stack elements after popping few // element/s
    System.out.println("s1 after pop :\n" + s1); stack<String> s2
    = new stack<>(3);
    // Pushing elements to string stack - s2
```

```
// Element 1 - hello
s2.push("hello"); //
Element 2 - world
s2.push("world");
// Element 3 - java s2.push("java");
// Print string stack after pushing above string // elements
System.out.println("\ns2 after pushing 3
elements :\n" + s2);
System.out.println("s2 after pushing 4th element
:");

Declaring objects of Integer type
stack<Float> s3 = new stack<>(2);
// Pushing elements to float stack - s3
// Element 1 - 100.0
s3.push(100.0f); //
Element 2 - 200.0
s3.push(200.0f);
// Print string stack after pushing above float
// elements
System.out.println("\ns3 after pushing 2
```

```
elements :\n" + s3); // Print and display  
top element of stack s3  
System.out.println("top element of s3:\n"+ s3.top());  
}  
}
```

### Program no: 27

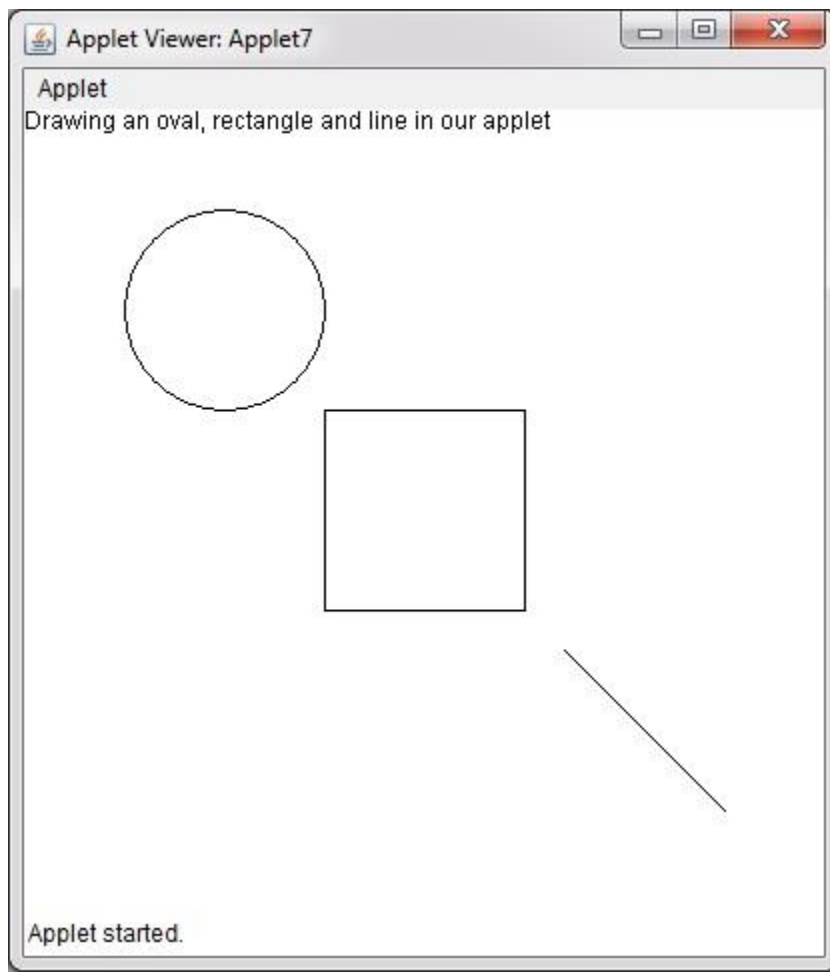
#### Figures

AIM: Program to draw Circle, Rectangle, Line in Applet

Program:

```
import java.applet.*;  
  
import java.awt.Graphics; public class figures extends  
Applet  
{  
    public void paint(Graphics g)  
    {  
        g.drawLine(30,30,300,30);  
        g.drawOval(100,100,100,100);  
        g.drawRect(250, 250, 200, 100);  
    }  
}
```

## Output:



Program no: 28

Numbers

AIM: Program to find maximum of three numbers using AWT.

Program:

```
import java.awt.*;  
import java.awt.event.*;  
import java.applet.*;  
public class largest extends Applet implements
```

```
ActionListener {  
  
    int a, b, c, result;  
  
    String str;  
  
    TextField Txt1 = new TextField(10);  
  
    TextField Txt2 = new TextField(10);  
  
    TextField Txt3 = new TextField(10);  
  
    TextField t4 = new TextField(10);  
  
    Label l2 = new Label("enter number 1: ");  
  
    Label l3 = new Label("enter number 2: ");  
  
    Label l5 = new Label("enter number 3: ");  
  
    Label l4 = new Label("largest : ");  
  
    Button b1 = new Button("click");  
  
    public void init() { add(l2);  
  
        add(Txt1); add(l3);  
  
        add(Txt2); add(l5);  
  
        add(Txt3); add(b1);  
  
        add(l4); add(t4);  
  
        b1.addActionListener(this);  
  
    }
```



```
public void actionPerformed(ActionEvent e) { if
(e.getSource() == b1)
{
str = Txt1.getText(); a =
Integer.parseInt(str); str
= Txt2.getText(); b =
Integer.parseInt(str); str
= Txt3.getText(); c =
Integer.parseInt(str); if
(a >= b && a >= c) {
result = a;
t4.setText(String.valueO
f(a)); repaint(); } else if
(b >= a && b >= c) {
result = b;
t4.setText(String.valueO
f(b)); repaint(); } else {
result = c;
t4.setText(String.valueO
f(c)); repaint();
```

```
}
```

```
}
```

```
}
```

```
}
```

### Program no: 29

#### STudents

AIM: Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise..

Program:

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.applet.*;
```

```
public class myline extends Applet implements
```

```
ActionListener { private int SMILE = 0; private float k; int i;
```

```
float j;
```

```
TextField T1 = new TextField(10);
```

```
TextField T2 = new TextField(10);
```

```
TextField t3 = new TextField(10);
```

```
Label l2 = new Label("enter total marks obtained : ");
```

```
Label l3 = new Label("enter total Marks : ");
```

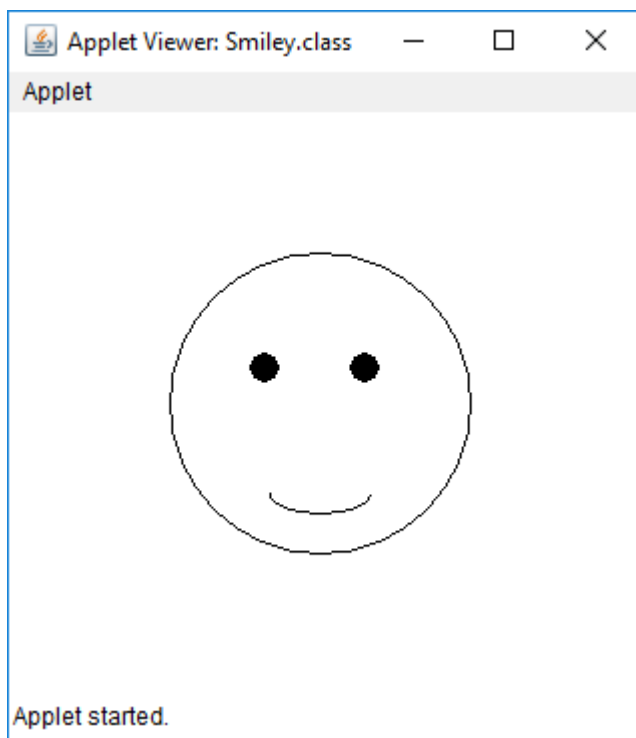
```
Label l4 = new Label("percentage : ");
Button b = new Button("percentage");

public void init()
{ add(l2);
  add(T1); add(l3);
  add(T2); add(l4);
  add(t3); add(b);
  b.addActionListener(this);
}

public void actionPerformed(ActionEvent e)
{ if (e.getSource() == b) i =
  Integer.parseInt(T1.getText()); j =
  Integer.parseInt(T2.getText());
  k = i / j;
  k = k * 100;
  if (k >= 50) {
    SMILE = 1;
  } else {
    SMILE = 0;
  }
}
```

```
t3.setText(String.valueOf(k) + " %"); repaint();  
}  
  
public void paint(Graphics g) { g.drawOval(80, 70, 150, 150);  
g.setColor(Color.black);  
g.fillOval(120, 120, 15, 15);  
g.fillOval(170, 120, 15, 15);  
  
if (SMILE == 1) {  
g.drawArc(130, 180, 50, 20, 180, 180); SMILE = 0;  
} else {  
g.drawArc(130, 180, 50, 20, 180, -180); } }  
}
```

### Output:



## Program no: 30

Students

AIM: Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

Program:

```
import java.applet.*; import
java.awt.*;

import java.awt.event.MouseEvent; import
java.awt.event.MouseListener;

public class house extends Applet implements
MouseListener, Runnable { private Color door = Color.blue;
public void paint(Graphics g) { int x[] = { 150, 300, 225 };

int y[] = { 150, 150, 25 };

g.setColor(Color.orange);

g.fillRect(150, 150, 150, 200);

g.drawRect(150, 150, 150, 200);

g.setColor(door);

g.fillRect(200, 200, 50, 150); g.drawRect(200, 200, 50, 150);
g.setColor(Color.red);

g.fillPolygon(x, y, 3);
```

```
g.drawPolygon(x, y, 3);  
}  
  
public void init() {  
    this.setSize(200, 200);  
    addMouseListener(this); }  
  
public void run() {  
    while (true) {  
        repaint();  
        try {  
            Thread.sleep(5);  
        }  
        catch (InterruptedException e) { e.printStackTrace();  
        }  
    }  
}  
  
public void mouseClicked(MouseEvent e) { int x = e.getX(), y  
    = e.getY();  
    if (x <= 300) door =  
    Color.red; else door =  
    Color.blue; repaint();
```

```
}  
  
public void mousePressed(MouseEvent e) { }  
public void mouseReleased(MouseEvent e) { }  
public void mouseEntered(MouseEvent e) { }  
public void mouseExited(MouseEvent e) { }  
  
}
```

Output:

