

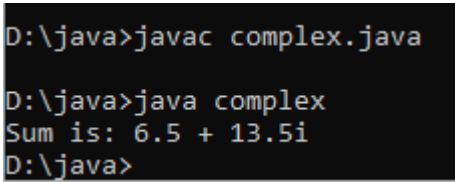
**OBJECT ORIENTED PROGRAMMING LAB
RECORD**

**Submitted By,
Ashtami Prasad
S2RMCA Batch A
RollNo:29**

1. Add complex numbers

```
public class Complex{  
    double real, img;  
    Complex(double r, double i){  
        this.real = r;  
        this.img= i;  
    }  
    public static Complex sum(Complex c1, Complex c2)  
    {  
        Complex temp = new Complex(0, 0);  
        temp.real = c1. real+ c2.real;  
        temp.img = c1.img+ c2.img;  
        return temp;  
    }  
    public static void main(String args[]) {  
        Complex c1 = new Complex(2, 10);  
        Complex c2 = new Complex(4.5, 3.5);  
        Complex temp = sum(c1, c2);  
        System.out.printf("Sum is: "+ temp.real+" " + temp.img +"i");  
    }  
}
```

Output



```
D:\java>javac complex.java  
D:\java>java complex  
Sum is: 6.5 + 13.5i  
D:\java>
```

2.define a class product with data members pcode,pname,price.create 3 objects of the class and find the product having the lowest price

```
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=2000;  
        p1.pname="laptop";  
        p1.price=10000;  
        p2.pcode=1110;  
        p2.pname="hp";  
        p2.price=35000;  
        p3.pcode=2002;  
        p3.pname="intel i3";  
        p3.price=40000;  
        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;  
            }  
        }  
    }  
}
```

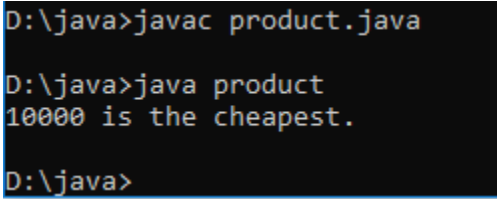
```
}
```

```
System.out.println(smallest + " is the cheapest.");
```

```
}
```

```
}
```

output



```
D:\java>javac product.java
```

```
D:\java>java product  
10000 is the cheapest.
```

```
D:\java>
```

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

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

```
public class mat {
```

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

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

```
System.out.println("Enter the number of row: ");
```

```
int row=ip.nextInt();
```

```
System.out.println("Enter the number of coloumn: ");
```

```
int col=ip.nextInt();
```

```
if(row==col)
```

```
{
```

```
System.out.println("Matrix is symmetric ");
```

```
}
```

```
else
```

```
System.out.println("Matrix is not symmetric ");
```

```
}
```

```
}
```

Output:

```
D:\java>javac mat.java

D:\java>java mat
Enter the number of row:
3
Enter the number of coloumn:
2
Matrix is not symmetric
```

4.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.

```
public class Cpu {
    int price;
    Cpu(int p) {
        this.price = p;
    }
    class Processor {
        int cores;
        String manufacture; Processor(int n, String m) {
            this.cores = n;
            this.manufacture = m;
        }
        void display() {
            System.out.println("No of Cores : " + this.cores);
            System.out.println("Processor manufactures : " + this.manufacture);
        }
    }
    static class Ram {
        int memory;
        String manufacture;
        Ram(int n, String m) {
            this.memory = n;
            this.manufacture = m;
        }
    }
}
```

```

void display() {
    System.out.println("Memory Size : " + this.memory);
    System.out.println("Memory manufactures : " + this.manufacture);
}

}

void display() {
    System.out.println("Price of CPU : " + this.price);
}

public static void main(String[] args) {
    Cpu intel = new Cpu(30000);
    Cpu.Processori_processor = intel.newProcessor(7, "intel");
    Cpu.Rami_ram = new Ram(1030, "hp");
    intel.display();
    i_processor.display();
    i_ram.display();
}
}

```



5: Area of different shapes using overloaded functions

```

public class ShapeA {
    int area(int side)
    {
        return side*side;
    }
    int area(int l,int b)
    {
        return l*b;
    }
}

```

```

    }

    double area(double b,double h)
    {
        return (0.5*(b*h));
    }

    double area(double r)
    {
        return (3.14*r*r);
    }

    public static void main(String[] args)
    {
        ShapeAobj=new ShapeA();
        System.out.println("Area of Square: "+obj.area(5));
        System.out.println("Area of Rectangle: "+obj.area(5,4));
        System.out.println("Area of Triangle:"+obj.area(5.5,2.1));
        System.out.println("Area of Circle: "+obj.area(5.7));
    }
}

```

Output:

```

D:\java>javac shape.java

D:\java>java shape
Area of Square: 4
Area of Rectangle: 8
Area of cone: 23.55
Area of Circle: 16.610599999999998
Area of cylinder: 34.540000000000006

```

6: 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.

```
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("Employee id : "+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 Employee id 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("Teacher's List is \n");

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

    obj[i].display();

}

}

```

Output:

```

Microsoft Windows [Version 10.0.17134.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\hp>d:

D:\>cd java

D:\java>javac Teacher.java

D:\java>java Teacher
Enter number of Teachers : 
1
Enter Employee id of teacher 1 : 101
Enter Name of teacher 1 : anu
Enter Salary of teacher 1 : 50000
Enter Address of teacher 1 : anubhavan
Enter department of teacher 1 : mca
Enter Subjects of teacher 1 : java
Teacher's List is

Employee id : 101 Name : anu Salary : 50000.0 Address : anubhavan department : mca Subjects : java

```

7: 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.

```

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 are");
    }
}
```

```

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

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

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 are");

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,dept1,tid1);
}


for(int i=0;i<n;i++) {
obj[i].display();
}
}

}
```

Output:

```

D:\java>javac Main.java
D:\java>java Main
Enter number of Teachers :
1
Enter the person name:
anu
Enter the Gender:
female
Enter the Address:
anubhavan
Enter the Age:
25
Enter the Employee id:
203
Enter the Company name:
evosoft
Enter the Salary:
25000
Enter the Qualification:
mca
Enter the Teacher id:
102
Enter the Department:
mca
Enter the Subject:
java
Personal details are
Name : anu Gender : female Age :25
Employee details are
Empid : 203 company_name : evosoft Salary : 25000.0 Address : anubhavan qualification : mca
Teacher's details are
teacherid : 102 department : mca Subjects : java

```

8: 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.

```

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:

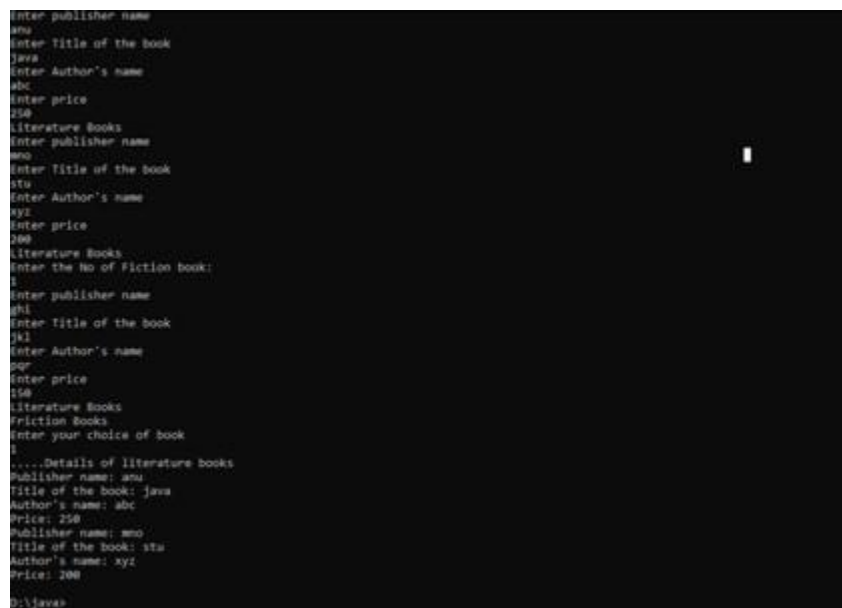
System.out.println("Wrong input");

}

}

}

```



```

Enter publisher name
anu
Enter Title of the book
java
Enter Author's name
abc
Enter price
250
Literature Books
Enter publisher name
ano
Enter Title of the book
stu
Enter Author's name
xyz
Enter price
200
Literature Books
Enter the No of Fiction book:
1
Enter publisher name
pqr
Enter Title of the book
kl1
Enter Author's name
pqr
Enter price
150
Literature Books
Fiction Books
Enter your choice of book
1
.....Details of literature books
Publisher name: anu
Title of the book: java
Author's name: abc
Price: 250
Publisher name: ano
Title of the book: stu
Author's name: xyz
Price: 200
D:\java>

```

9: Create classes Student and sports.Create another class result inherited from student and sports.Display the academic and sports score of a student.

```

interface student

{

void stresultt();

```

```
}  
  
interface sports  
{  
    void spresult();  
}  
  
class result implements student,sports{  
    public void spresult()  
    {  
        String eighthundred="First";  
        String twohundred="Second";  
        String longjump="First";  
        String relay="Second";  
        System.out.println("Sports Result");  
        System.out.println("eight hundredmerter:"+ eighthundred);  
        System.out.println("Two Hundred Meter:"+twohundred);  
        System.out.println("long jump:"+longjump);  
        System.out.println("Relay:"+relay);  
    }  
    public void stresultt()  
    {  
        int physics=50;  
        int chemistry=60;  
        int biology=40;  
        int hindi=40;  
        int social=77;  
        System.out.println("Marks");  
        System.out.println("physics:"+physics);  
        System.out.println("chemistry:"+chemistry);  
        System.out.println("biology:"+biology);  
        System.out.println("hindi:"+hindi);  
        System.out.println("social:"+social);  
    }  
}
```

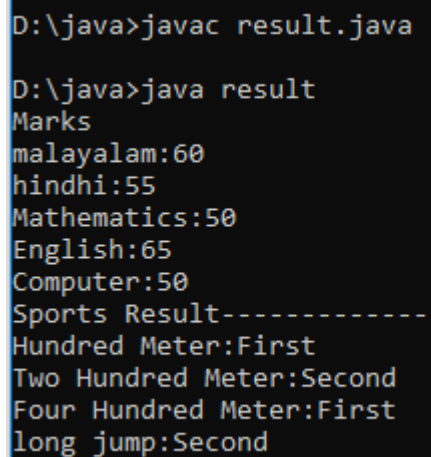
```

}

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

```

Output:



```

D:\java>javac result.java

D:\java>java result
Marks
malayalam:60
hindhi:55
Mathematics:50
English:65
Computer:50
Sports Result-----
Hundred Meter:First
Two Hundred Meter:Second
Four Hundred Meter:First
long jump:Second

```

10:Create an interface having prototype 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.

```

import java.util.Scanner;

interface Shape
{
    void input();
    void area();
    void perimeter();
}

class Circle implements Shape
{
    int r = 0;
    double pi = 3.14, ar = 0, per=0;
    public void input()

```

```
{ Scanner s = new Scanner(System.in); System.out.print("Enter radius of circle:");  
r= s.nextInt();  
}  
public void area()  
{  
ar = pi * r * r;  
System.out.println("Area of circle:"+ar);  
}  
public void perimeter()  
{  
per = 2 * pi * r;  
System.out.println("Perimeter of circle:"+per);  
}  
}  
class Rectangle implements Shape  
{  
int l = 0, b = 0;  
double ar,per;  
public void input()  
{ Scanner s = new Scanner(System.in);  
System.out.print("Enter length of rectangle:");  
l = s.nextInt();  
System.out.print("Enter breadth of rectangle:");  
b = s.nextInt();  
}  
public void area()  
{  
ar = l * b;  
System.out.println("Area of rectangle:"+ar);  
}  
public void perimeter()
```

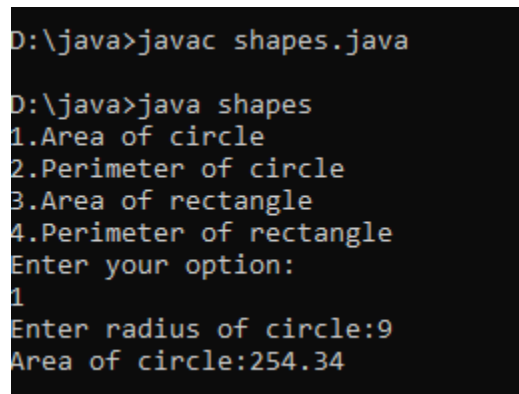
```
{  
    per = 2 * (l + b);  
    System.out.println("Perimeter of rectangle:"+per);  
}  
}  
  
public class shapes  
{  
    public static void main(String[] args)  
    { int n;  
      Scanner s = new Scanner(System.in);  
      Rectangle obj1 = new Rectangle();  
      Circle obj2 = new Circle(); System.out.println("1.Area of circle");  
      System.out.println("2.Perimeter of circle");  
      System.out.println("3.Area of rectangle");  
      System.out.println("4.Perimeter of rectangle");  
      System.out.println("Enter your option:");  
      n= s.nextInt();  
      switch(n) {  
      case 1:  
          obj2.input();  
          obj2.area();  
          break;  
      case 2:  
          obj2.input();  
          obj2.perimeter();  
          break;  
      case 3:  
          obj2.input();  
          obj2.area();  
          break;  
      case 4:
```

```

obj2.input();
obj2.perimeter();
break;
default:
System.out.println("Invalid option");
}
}
}

```

Output:



```

D:\java>javac shapes.java
D:\java>java shapes
1.Area of circle
2.Perimeter of circle
3.Area of rectangle
4.Perimeter of rectangle
Enter your option:
1
Enter radius of circle:9
Area of circle:254.34

```

10:Prepare bill with the given format using calculate method from interface.

Order No.Date Productid name quantity price total

101 A 2 25 50 102 B 1 100 100

Net.Amount 150

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);
}
}

```

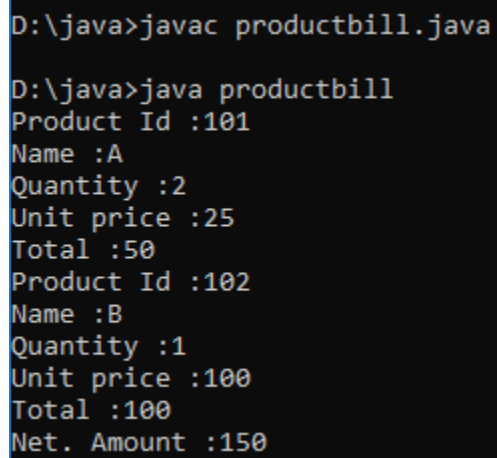
```
System.out.println("Name :"+name);
System.out.println("Quantity :"+quantity);
System.out.println("Unit price :"+unit);
System.out.println("Total :"+total);
return(total);
}
}

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:



```
D:\java>javac productbill.java  
  
D:\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
```

11. program to sort strings

```
public class sortstring{  
    public static void main(String[] args)  
    {  
        String names[]={"hai","hello","how","are","you"};  
        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(" sorted array of string is :");

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

{

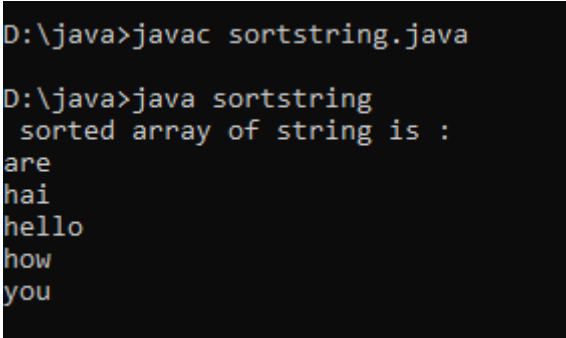
System.out.println(names[i]);

}

}

}

```



```

D:\java>javac sortstring.java

D:\java>java sortstring
sorted array of string is :
are
hai
hello
how
you

```

12. search an element in an array

```

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 want to search :");

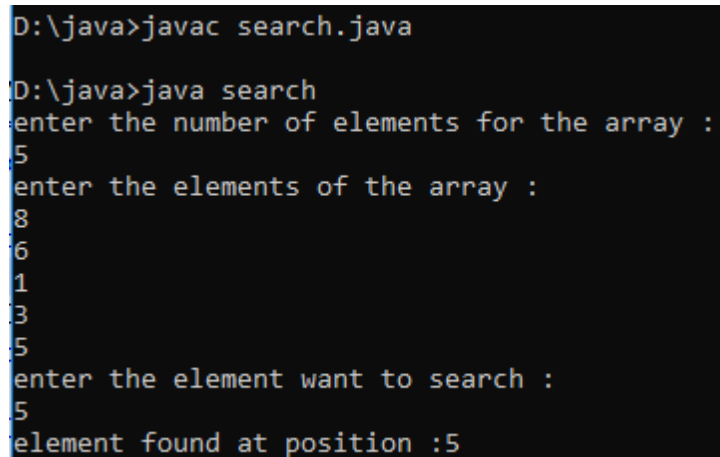
b=s.nextInt();

```

```

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");
}
}
}

```



```

D:\java>javac search.java
D:\java>java search
enter the number of elements for the array :
5
enter the elements of the array :
8
6
1
3
5
enter the element want to search :
5
element found at position :5

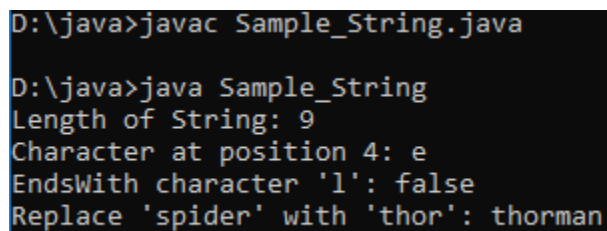
```

13. perform string manipulations

```

public class Sample_String{
    public static void main(String[] args){
        String str_Sample = "spiderman";
        System.out.println("Length of String: " + str_Sample.length());
        System.out.println("Character at position 4: " + str_Sample.charAt(4));
        System.out.println("EndsWith character 'l': " + str_Sample.endsWith("l"));
        System.out.println("Replace 'spider' with 'thor': " + str_Sample.replace("spider", "thor"));
    }
}

```



```

D:\java>javac Sample_String.java
D:\java>java Sample_String
Length of String: 9
Character at position 4: e
EndsWith character 'l': false
Replace 'spider' with 'thor': thorman

```

14:..Java program to create generic stack and do the push and pop operation

A stack class is provided by the Java collection framework and it implements the Stack data structure. The stack implements LIFO i.e. Last In First Out. This means that the elements pushed last are the ones that are popped first.

1. push() Method adds element x to the stack.
2. pop() Method removes the last element of the stack.
3. top() Method returns the last element of the stack.
4. empty() Method returns whether the stack is empty or not.

```

import java.io.*;
import java.util.*;

public class Example {
    public static void main (String[] args) {
        Stack<Integer> s = new Stack<Integer>();
        s.push(5);
        s.push(1);
    }
}

```

```

s.push(9);
s.push(4);
s.push(8);
System.out.print("The stack is: " + s);
System.out.print("\nThe element popped is: ");
Integer num1 = (Integer) s.pop();
System.out.print(num1);
System.out.print("\nThe stack after pop is: " + s);
Integer pos = (Integer) s.search(9);
if(pos == -1)
System.out.print("\nThe element 9 not found in stack");
else
System.out.print("\nThe element 9 is found at position " + pos + " in stack");
}
}

```

output

The stack is: [5, 1, 9, 4, 8]

The element popped is: 8

The stack after pop is: [5, 1, 9, 4]

The element 9 is found at position 2 in stack

15:Generic method implement bubble sort

Bubble sort is a simple sorting algorithm. This sorting algorithm is a comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large datasets as its average and worst case complexity is of $O(n^2)$ where n is the number of items.

```

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[] = { 2, 5, -2, 6, -3, 8, 0, -7, -9, 4 };

    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

```
D:\java>javac bubblesort.java
```

```
D:\java>java bubblesort
Array Before Bubble Sort
2 5 -2 6 -3 8 0 -7 -9 4
Array After Bubble Sort
-9 -7 -3 -2 0 2 4 5 6 8
```

16. Maintain a list of string using arraylist from a collection of framework, perform builtin operation

The ArrayList class extends AbstractList and implements the List interface. ArrayList supports dynamic arrays that can grow as needed.

Standard Java arrays are of a fixed length. After arrays are created, they cannot grow or shrink, which means that you must know in advance how many elements an array will hold.

Array lists are created with an initial size. When this size is exceeded, the collection is automatically enlarged. When objects are removed, the array may be shrunk.

```
import java.util.*;

public class ArrayListDemo {

    public static void main(String args[]) {

        // create an array list
        ArrayList al = new ArrayList();

        System.out.println("Initial size of al: " + al.size());

        // add elements to the array list
        al.add("C");
        al.add("A");
        al.add("E");
        al.add("B");
        al.add("D");
        al.add("F");
        al.add(1, "A2");

        System.out.println("Size of al after additions: " + al.size());
        System.out.println("Contents of al: " + al);

        al.remove("F");
        al.remove(2);
```

```
System.out.println("Size of al after deletions: " + al.size());

System.out.println("Contents of al: " + al);

}

}
```

Output

```
D:\java>java ArrayListDemo
Initial size of al: 0
Size of al after additions: 7
Contents of al: [C, A2, A, E, B, D, F]
Size of al after deletions: 5
Contents of al: [C, A2, E, B, D]
```

17: Write a user defined exception class to authentication the user name and password.

```
import java.util.Scanner;

class UsernameException extends Exception {

    public UsernameException(String msg) {

        super(msg);

    }

}

class PasswordException extends Exception {

    public PasswordException(String msg) {

        super(msg);

    }

}

public class checkLogin {

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);

        String username, password;

        System.out.print("Enter username :: ");

        username = s.nextLine();

        System.out.print("Enter password :: ");
```

```

password = s.nextLine();

int length = username.length();

try {
if(length < 6)

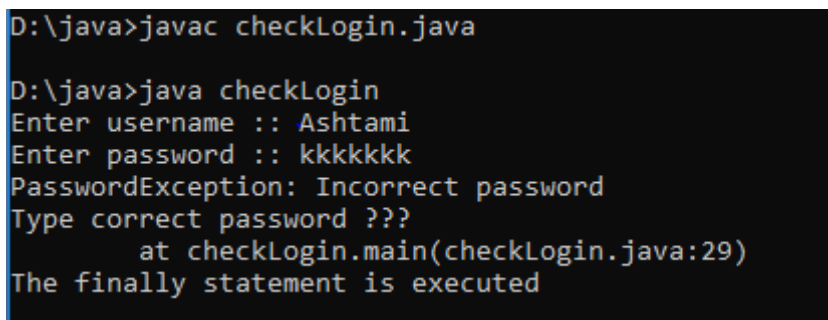
throw new UsernameException("Username must be greater than 6 characters ???");
else if(!password.equals("hello"))
throw new PasswordException("Incorrect password\nType correct password ???");
else
System.out.println("Login Successful !!!");
}

catch (UsernameException u) {
u.printStackTrace();
}

catch (PasswordException p) {
p.printStackTrace();
}

finally {
System.out.println("The finally statement is executed");
}
}
}

```



```

D:\java>javac checkLogin.java

D:\java>java checkLogin
Enter username :: Ashtami
Enter password :: kkkkkkk
PasswordException: Incorrect password
Type correct password ???
    at checkLogin.main(checkLogin.java:29)
The finally statement is executed

```

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


```
import java.util.Scanner;

import java.util.InputMismatchException;

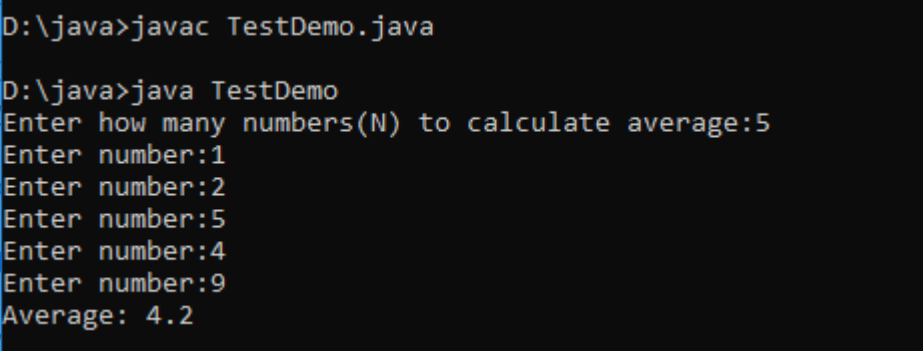
public class TestDemo
{
    public static void main(String args[])
    {
        double total = 0, N, userInput;
        Scanner input = new Scanner(System.in);
        while (true)
        {
            System.out.print("Enter how many numbers(N) to calculate average:");
            userInput = input.nextDouble();
            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);
}
}

```



```

D:\java>javac TestDemo.java
D:\java>java TestDemo
Enter how many numbers(N) to calculate average:5
Enter number:1
Enter number:2
Enter number:5
Enter number:4
Enter number:9
Average: 4.2

```

19: Define 2 classes one for generating multiplication table of 5 and other for displaying first N prime numbers implement using threads(thread class)

```

class ThreadA extends Thread{
    public void run( ) {
        int n = 5;
        for (int i = 1; i<= 10; ++i)
            System.out.println(n + " * " + i +
                " = " + n * i);
        System.out.println("Exiting from Thread A ...");
    }
}

class ThreadB extends Thread {
    public void run( ) {
        Scanner sc = new Scanner(System.in);
    }
}

```

```

int i,n,p,count,flag;

System.out.println("Enter the number of prime terms you want!");

n=sc.nextInt();

System.out.println("First "+n+" prime numbers are :-");

p=2;

i=1;

while(i<=n)

{

flag=1;

for(count=2;count<=p-1;count++)

{

if(p%count==0) //Will be true if p is not prime

{

flag=0;

break; //Loop will terminate if p is not prime

}

}

if(flag==1)

{

System.out.print(p+" ");

i++;

}

p++;

}

System.out.println("Exiting from Thread B ...");

}

}

public class Demonstration_111 {

public static void main(String args[]) {

ThreadA a = new ThreadA();

```

```
ThreadB b = new ThreadB();  
a.start();  
b.start();  
System.out.println("... Multithreading is over ");  
}  
}
```

20 : Define 2 classes one for generating fibanocci numbers and other for displaying even numbers in a given range.implement using threads(runnable interface)

```
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(){
```

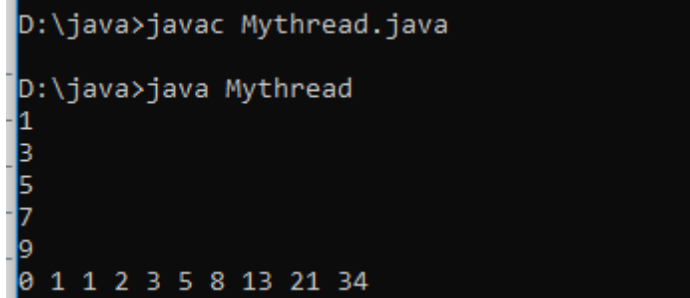
```

        int n1=0,n2=1,n3,i,count=10;
System.out.print(n1+" "+n2);

for(i=2;i<count;++i)
{
    n3=n1+n2;
    System.out.print(" "+n3);
    n1=n2;
    n2=n3;
}
}
}

```

Output:



```

D:\java>javac Mythread.java
D:\java>java Mythread
0
1
3
5
7
9
0 1 1 2 3 5 8 13 21 34

```

21: Program to draw circle,rectangle,line in applet

```

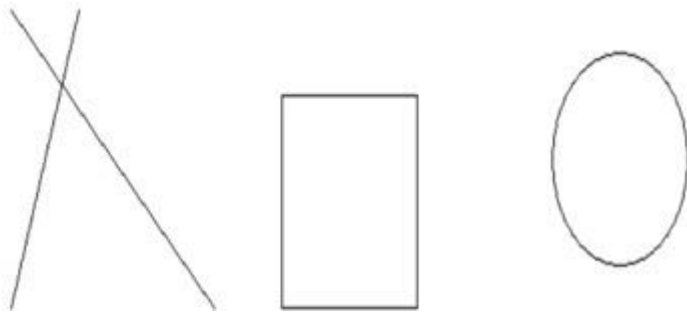
import java.awt.*;
import java.applet.*;
public class line extends Applet
{
    public void paint(Graphics g)
    {
        g.drawLine(100,10,250, 150);
        g.drawLine(100,150,150,10);
        g.setColor(Color.black);
    }
}

```

```
g.drawRect(300, 50, 100, 100);  
g.setColor(Color.black);  
g.drawOval(500,30,100,100);  
}  
}
```

.html code

```
<html>  
<head>  
</head>  
<body>  
<applet code = "line.class" width = "420" height = "320"></applet>  
</body>  
</html>
```



22: Program to find maximum of three numbers using AWT

```
import java.awt.*;  
import java.awt.Event;  
import java.applet.*;  
public class largest extends Applet  
{  
    TextField Txt1,Txt2,Txt3;  
    public void init(){  
        Txt1 = new TextField(10);  
        Txt2 = new TextField(10);  
        Txt3 = new TextField(10);
```

```
add(Txt1);
add(Txt2);
add(Txt3);
}

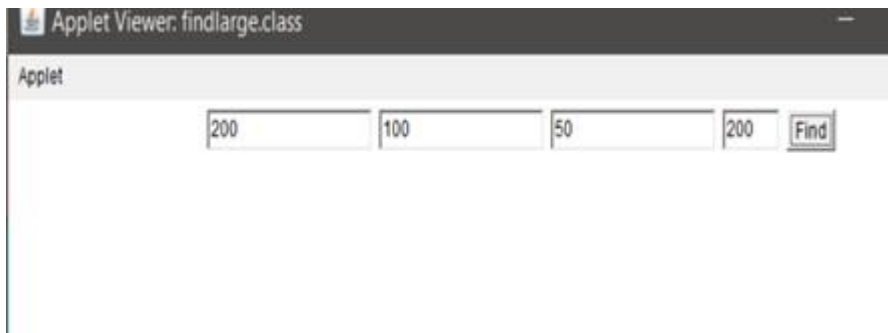
public void paint(Graphics g){
int a, b, c,result;
String str;
g.drawString("Enter the numbers ",15,15);
str=Txt1.getText();
a=Integer.parseInt(str);
str=Txt2.getText();
b=Integer.parseInt(str);
str=Txt3.getText();
c=Integer.parseInt(str);
if (a>=b && a>=c)
{
result=a;
}
else if(b>=a && b>=c)
{
result=b;
}
else
{
result=c;
}
g.drawString("Largest number is "+result,10,70);
}

public boolean action(Event e, Object o){
repaint();
return true;
}
```

```

}
}
html
<html>
<head>
</head>
<body>
<div align="center">
<applet code="largest.class"width="800"height="500">
</applet>
</div>
</body>
</html>

```



23: Find the percentage of marks obtained by a student in 5 subject. Display a happy face if he secures above

50% or a sad face if otherwise.

```

import java.awt.*;
import java.awt.event.*;
import java.applet.*;

public class marks extends Applet implements ActionListener {

    public int per = 0;

    Label l1 = new Label("enter Marks of Subject 1: ");
    Label l2 = new Label("enter Marks of Subject 2: ");

```



```
Label l3 = new Label("enter Marks of Subject 3: ");
Label l4 = new Label("enter Marks of Subject 4: ");
Label l5 = new Label("enter Marks of Subject 5: ");
Label l6 = new Label("Total Percentage: ");
TextField t1 = new TextField(10);
TextField t2 = new TextField(10);
TextField t3 = new TextField(10);
TextField t4 = new TextField(10);
TextField t5 = new TextField(10);
TextField t6 = new TextField(10);
Button b1 = new Button("CALCULATE PERCENTAGE");
public marks()
{
l1.setBounds(50, 100, 280, 20);
l2.setBounds(50, 150, 280, 20);
l3.setBounds(50, 200, 280, 20);
l4.setBounds(50, 250, 280, 20);
l5.setBounds(50, 300, 280, 20);
l6.setBounds(50, 350, 280, 20);
t1.setBounds(200, 100, 300, 20);
t2.setBounds(200, 150, 300, 20);
t3.setBounds(200, 200, 300, 20);
t4.setBounds(200, 250, 300, 20);
t5.setBounds(200, 300, 300, 20);
t6.setBounds(200, 350, 300, 20);
b1.setBounds(200,400, 200, 20);
GridLayout g1 = new GridLayout(20, 2, 5, 5);
setLayout(g1);
add(l1);
add(t1);
add(l2);
```

```
add(t2);
add(l3);
add(t3);
add(l4);
add(t4);
add(l5);
add(t5);
add(l6);
add(t6);
add(b1);
b1.addActionListener(this);
}

@Override
public void actionPerformed(ActionEvent e) {
    int m1 = Integer.parseInt(t1.getText());
    int m2= Integer.parseInt(t2.getText());
    int m3= Integer.parseInt(t3.getText());
    int m4= Integer.parseInt(t4.getText());
    int m5= Integer.parseInt(t5.getText());
    if(e.getSource()==b1)
    {
        int add=m1+m2+m3+m4+m5;
        per=add/5;
        t6.setText(String.valueOf(per)+" %");
        repaint();
    }
}

public void paint(Graphics g)
{
    if(per>=50)
    {
```

```

g.setColor(Color.yellow);
g.drawOval(80, 700, 150, 150);
g.fillOval(80, 700, 150, 150);
g.setColor(Color.BLACK);
g.fillOval(120, 740, 15, 15);
g.fillOval(170, 740, 15, 15);
g.drawArc(130, 800, 50, 20, 180, 180);
}
else if(per>0 && per<50)
{
g.setColor(Color.yellow);
g.drawOval(80, 700, 150, 150);
g.fillOval(80, 700, 150, 150);
g.setColor(Color.BLACK);
g.fillOval(120, 740, 15, 15);
g.fillOval(170, 740, 15, 15);
g.drawArc(130,820,50,20,0,180);
}
}

public static void main(String args[]) {
new marks();
}
}

Html
<html>
<head>
</head>
<body>
<div align="center">
<applet code="marks.class"width="800"height="500">
</applet>

```

</div>

</body>

</html>

10. Percentage Worksheet	
Applet	
enter Marks of Subject 1:	
44	
enter Marks of Subject 2:	
45	
enter Marks of Subject 3:	
46	
enter Marks of Subject 4:	
47	
enter Marks of Subject 5:	
48	
Total Percentage	
49 %	
CALCULATE PERCENTAGE	



Applet started

24:Using 2D graphics commands in an applet ,construct a house .On mouse click event change the color of the

door from blue to red.

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

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

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

```
public class house extends Applet implements MouseListener, Runnable {
```

```
    private Color doorColor = Color.WHITE;
```

```
    public void paint(Graphics gp) {
```

```
        int[] i = { 150, 300, 225 };
```

```
        int[] j = { 150, 150, 25 };
```

```
        gp.drawRect(150, 150, 150, 200);
```

```
        gp.drawOval(200, 75, 50, 50);
```

```
        gp.drawPolygon(i, j, 3);
```

```
        gp.setColor(doorColor);
```

```
gp.fillRect(200, 200, 50, 150);
gp.setColor(Color.BLACK);
gp.drawRect(200, 200, 50, 150);
}

public void init() {
this.setSize(200, 200);
addMouseListener(this);
}

public void run() {
while (true) {
repaint();
try {
Thread.sleep(17);
} catch (InterruptedException e) {
e.printStackTrace();
}
}
}

public void mouseClicked(MouseEvent e) {
int x = e.getX(), y = e.getY();
if (x >= 200 && x <= 250 && y >= 200 && y <= 350)
doorColor = Color.RED;
else
doorColor = Color.BLUE;
repaint();
System.out.println("Mouse Position : X=" + x + " Y=" + y + "");
}

public void mousePressed(MouseEvent e) {
}

public void mouseReleased(MouseEvent e) {
}
```

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

Html code

```
<html>  
<head>  
</head>  
<body>  
<div align="center">  
<applet code="house.class" width="800" height="500">  
</applet>  
</div>  
</body>  
</html>
```





25: Implement a simple calculator using AWT components

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

public class calc extends Applet implements ActionListener {

    Frame f = new Frame();

    Label l1 = new Label("enter number");
    Label l2 = new Label("enter number");
    Label l3 = new Label("result");

    TextField t1 = new TextField(10);
    TextField t2 = new TextField(10);
    TextField t3 = new TextField(10);

    Button b1 = new Button("ADD");
    Button b2 = new Button("SUB");
    Button b3 = new Button("MUL");
    Button b4 = new Button("DIV");

    calc() {

        l1.setBounds(50, 100, 100, 20);
        l2.setBounds(50, 100, 100, 20);
        l3.setBounds(50, 100, 100, 20);

        t1.setBounds(200, 100, 100, 20);
        t2.setBounds(250, 150, 100, 20);
        t3.setBounds(300, 200, 100, 20);
```

```
b1.setBounds(50, 250, 50, 20);
b2.setBounds(110, 250, 50, 20);
b3.setBounds(170, 250, 50, 20);
b4.setBounds(230, 250, 50, 20);
f.add(l1);
f.add(t1);
f.add(l2);
f.add(t2);
f.add(l3);
f.add(t3);
f.add(b1);
f.add(b2);
f.add(b3);
f.add(b4);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
f.setLayout(null);
f.setVisible(true);
f.setSize(500, 500);
}

public void actionPerformed(ActionEvent e) {
    int i = Integer.parseInt(t1.getText());
    int j = Integer.parseInt(t2.getText());
    if (e.getSource() == b1) {
        t3.setText(String.valueOf(i + j));
    }
    if (e.getSource() == b2) {
        t3.setText(String.valueOf(i - j));
    }
}
```



```

if (e.getSource() == b3) {
t3.setText(String.valueOf(i * j));
}
if (e.getSource() == b4) {
t3.setText(String.valueOf(i / j));
}
}
public static void main(String args[]) {
new calc();
}
}

```

enter number

26: Develop a program that has a choice component which contains the names of shapes such as rectangle

,triangle.square and circle,Draw the corresponding shapes for given parameters as per user's choice.

```

import java.applet.Applet;
import java.awt.*;

```

```
import java.awt.Graphics;

import java.awt.event.*;

public class figchoice extends Applet implements ItemListener {

    Choice ch;

    int x1[] = {50,120,220,20};

    int y1[] = {50,120,20,20};

    int n=4;

    int Selection;

    public void init()

    {

        ch = new Choice();

        ch.addItem("Select a Shape");

        ch.addItem("Rectangle");

        ch.addItem("Triangle");

        ch.addItem("Square");

        ch.addItem("Circle");

        add(ch);

        ch.addItemListener(this);

    }

    public void itemStateChanged (ItemEvent e)

    {

        Selection = ch.getSelectedIndex();

        repaint();

    }

    public void paint(Graphics g)

    {

        super.paint(g);

        if (Selection == 1)

        {

            g.drawRect(50,50,100,150);

        }

    }

}
```

```
if (Selection == 2)
{
g.drawPolygon(x1,y1,n);
}
if (Selection == 3)
{
g.drawRect(50,50,100,100);
}
if (Selection == 4)
{
g.drawOval(70,30,100,100);
}
```

Applet viewer: tgcchoice.class

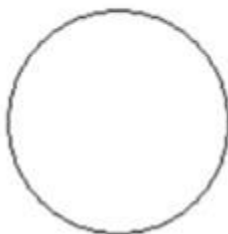
Applet

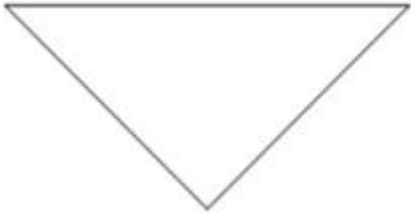
Rectangle ▾



Applet

Circle ▾





Triangle ▾

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

```
import java.util.*;

class arrayjava{

    public static void main(String args[]){
        ArrayList<String>alist=new ArrayList<String>();
        alist.add("appu");
        alist.add("ammu");
        alist.add("minnu");
        alist.add("thomu");
        alist.add("pinky");
        alist.add("Tom");

        System.out.println(alist);

        alist.add(3, "appu");

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

```
D:\java>javac arrayjava.java

D:\java>java arrayjava
[appu, ammu, minnu, thomu, pinky, Tom]
[appu, ammu, minnu, appu, thomu, pinky, Tom]
```

28.Program to remove all the elements from a linked list

```
import java.util.*;

public class removelink {

    public static void main(String[] args) {

        // create an empty linked list
        LinkedList<String>l_list = new LinkedList<String>();

        // use add() method to add values in the linked list
        l_list.add("violet");
        l_list.add("Green");
        l_list.add("Black");
        l_list.add("Pink");
        l_list.add("blue");


        // print the list
        System.out.println("The Original linked list: " + l_list);

        // Removing all the elements from the linked list
        l_list.clear();

        System.out.println("The New linked list: " + l_list);

    }

}
```

```
D:\java>javac removelink.java

D:\java>java removelink
The Original linked list: [violet, Green, Black, Pink, blue]
The New linked list: []
```

29.program to demonstrate the addition and deletion of elements in dequeue

```
import java.util.*;

public class DequeueExample {
```

```

public static void main(String[] args)
{
    Deque<String> deque
= new LinkedList<String>();
    // We can add elements to the queue
    // in various ways
    // Add at the last
    deque.add("Element 1 (Tail)");
    // Add at the first
    deque.addFirst("Element 2 (Head)");
    // Add at the last
    deque.addLast("Element 3 (Tail)");
    // Add at the first
    deque.push("Element 4 (Head)");
    // Add at the last
    deque.offer("Element 5 (Tail)");
    // Add at the first
    deque.offerFirst("Element 6 (Head)");
    System.out.println(deque + "\n");
    // We can remove the first element
    // or the last element.
    deque.removeFirst();
    deque.removeLast();
    System.out.println("Deque after removing "
+ "first and last: "
+ deque);
}
}

```

```

D:\java>java DequeExample
[Element 6 (Head), Element 4 (Head), Element 2 (Head), 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)]

```

30.program to demonstrate the working of map interface by adding ,removing,changing

```
import java.util.*;

class hashmap {

    public static void main(String args[])

    {

        Map<String, Integer> hm

        = new HashMap<String, Integer>();

        hm.put("a", new Integer(200));

        hm.put("b", new Integer(400));

        hm.put("c", new Integer(600));

        hm.put("d", new Integer(800));

        // Traversing through the map

        for (Map.Entry<String, Integer>me :hm.entrySet()) {

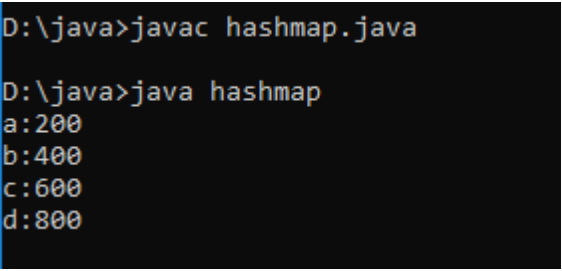
            System.out.print(me.getKey() + ":");

            System.out.println(me.getValue());

        }

    }

}
```



```
D:\java>javac hashmap.java

D:\java>java hashmap
a:200
b:400
c:600
d:800
```

31:program to convert hash map to tree map

```
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);

}

}

```

```

D:\java>javac HT.java

D:\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}

```

32:Program to list the sub directories and files in a given directory and also search for a file name

```

import java.io.File;

import java.util.*;

import java.io.*;

public class p1 {

    public static final String RED="\033[0;31m";

    public static final String RESET="\033[0m";

    static void RecursivePrint(File[] arr, int index, int level, String searchfor) {

        // exit condition

        if (index == arr.length)

            return;

        // space for internbal level

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

            System.out.print("\t");

        if(arr[index].getName().toLowerCase().contains(searchfor))

            System.out.print(RED);

        else

            System.out.print(RESET);
    }
}

```



```

// for files
if (arr[index].isFile())
System.out.println(arr[index].getName());
else if (arr[index].isDirectory()) {
System.out.println "[" + arr[index].getName() + ""];
RecursivePrint(arr[index].listFiles(), 0, level + 1, searchfor);
}

RecursivePrint(arr, ++index, level, searchfor);
}

public static void main(String[] args) {
Scanner scan = new Scanner(System.in);
System.out.println("Enter the directory path");
String maindirpath = scan.nextLine();
System.out.println("Enter the file/directory name to search");
String searchfor = scan.nextLine();
File maindir = new File(maindirpath);
if (maindir.exists() && maindir.isDirectory()) {
File arr[] = maindir.listFiles();
System.out.println("#####");
System.out.println("Files from main directory" + maindir);
System.out.println("#####");
RecursivePrint(arr, 0, 0, searchfor.toLowerCase()); // array,index
}
}
}

```

Output:

```

D:\java>javac p1.java
D:\java>java p1
Enter the directory path
D:\java
Enter the file/directory name to search
area.java
#####
Files from main directoryD:\java
#####
[0mAdditionOperation.java
[0mapplet.html
[0marea.class

```

33:Write a program to write to a file ,then read from the file and display the contents on the console

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;

class read {

    public static void main(String[] args) {
        String var = "";
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter the text to create file : type exit to stop"
        );
        while (!var.endsWith("exit\n"))
            var = var + scan.nextLine()+"\n";
        try {
            File file = new File("output.txt");
            FileWriter fw = new FileWriter(file);
            fw.write(var);
            fw.close();
            System.out.println("Reading File content");
            FileReader fr = new FileReader("output.txt");
            String str = "";
            int i;
            while ((i = fr.read()) != -1) {
                // Storing every character in the string
                str += (char) i;
            }
            System.out.println(str);
            fr.close();
        } catch (IOException e) {
            System.out.println("There are some exception");
        }
    }
}
```

Output:

```
D:\java>java read
Enter the text to create file : type exit to stop
program
desktop
exit
Reading File content
program
desktop
exit
```

34: Write a program to copy one file to another**Pre-requisite**

Create a text file with content where the java program is running for reading

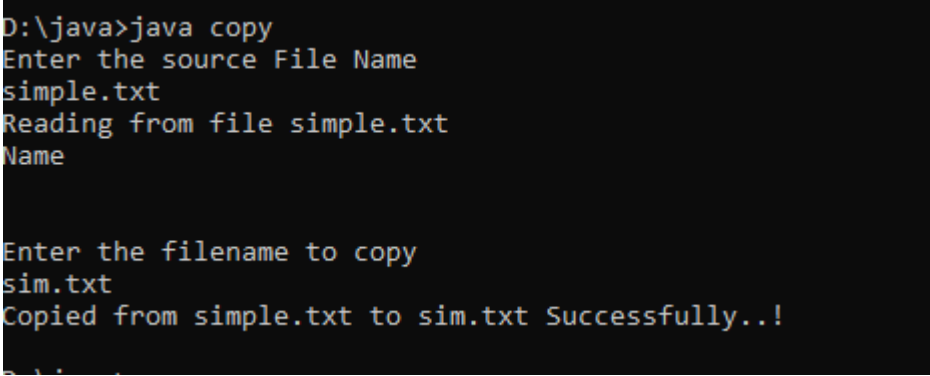
```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;

public class copy {
    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        System.out.println("Enter the source File Name");
        String source=scan.nextLine();
        try {
            FileReader fr=new FileReader(source);
            String str = "";
            int i;
            System.out.println("Reading from file "+source);
            while ((i = fr.read()) != -1) {
                // Storing every character in the string
                str += (char) i;
            }
            System.out.println(str);
            System.out.println("\nEnter the filename to copy");
            String destination=scan.nextLine();
            File file=new File(destination);
            FileWriter fw = new FileWriter(file);
```

```

fw.write(str);
fr.close();
fw.close();
System.out.println("Copied from "+source+" to "+destination+ " Successfully..!");
} catch (Exception e) {
//TODO: handle exception
System.out.println("Exception Occured");
}
}
}

```



```

D:\java>java copy
Enter the source File Name
simple.txt
Reading from file simple.txt
Name

Enter the filename to copy
sim.txt
Copied from simple.txt to sim.txt Successfully..!

```

35: Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files

Pre-requisite

Create a text file with content of numbers where the java program is running for reading numbers

```

import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;
public class oddeven {
    public static void main(String[] args) {
        try {
            FileReader fr = new FileReader("numbers.txt");
            BufferedReader br = new BufferedReader(fr);
            File file1 = new File("oddnumbers.txt");
            FileWriter fw1 = new FileWriter(file1);

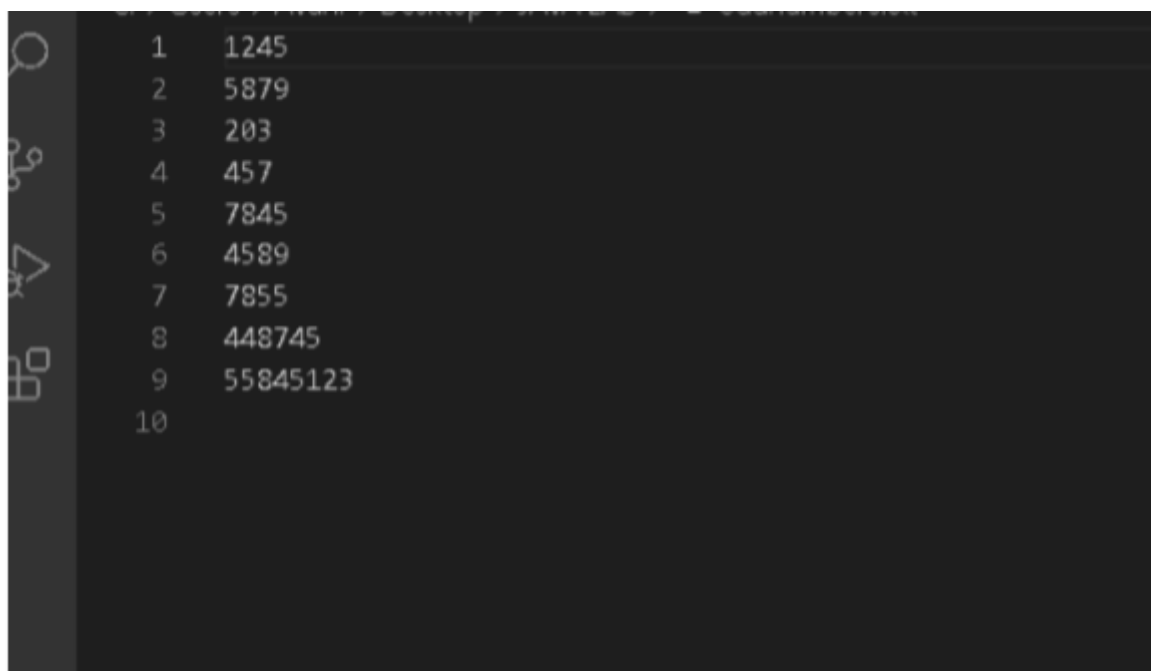
```

```
File file2 = new File("evennumbers.txt");
FileWriter fw2 = new FileWriter(file2);
String num;
while ((num = br.readLine()) != null) {
    if (Integer.parseInt(num) % 2 == 0) {
        fw2.write(num + "\n");
    } else {
        fw1.write(num + "\n");
    }
}
fw1.close();
fw2.close();
} catch (Exception e) {
    // TODO: handle exception
    System.out.println("Error");
}
}
```

Output:

```
D:\java>javac oddeven.java
```

```
D:\java>java oddeven
```

A screenshot of a Java application's output. It shows a list of numbers, each preceded by an index from 1 to 10. The numbers are: 1 1245, 2 5879, 3 203, 4 457, 5 7845, 6 4589, 7 7855, 8 448745, 9 55845123, and 10. The output is displayed in a dark-themed window with a sidebar on the left containing icons for search, file explorer, and other IDE features.

```
1 1245
2 5879
3 203
4 457
5 7845
6 4589
7 7855
8 448745
9 55845123
10
```

```
1 968758
2 3210
3 0
4 0124
5 784
6 55552132
7
```

```
1 1245
2 5879
3 203
4 457
5 7845
6 968758
7 4589
8 3210
9 0
10 0124
11 7855
12 784
13 448745
14 55845123
15 55552132
```

36.Client server communication using Socket – TCP/IP

PROGRAM

Server

```
import java.io.*;
import java.net.*;

public class MyServer {
    public static void main(String[] args) {
        try{
            ServerSocket ss=new ServerSocket(6666);
            Socket s=ss.accept(); //establishes connection
```

```
DataInputStream dis=new DataInputStream(s.getInputStream());  
String str=(String)dis.readUTF();  
System.out.println("message= "+str);  
ss.close();  
  
}  
catch(Exception e) { System.out.println(e);}  
}  
}
```

Output:

```
D:\java>javac MyClient.java
```

```
D:\java>java MyClient
```

```
D:\java>javac MyServer.java
```

```
D:\java>java MyServer
```