INDEX

	ii. Character at second and fourth position iii. Find the sub string using start index only iv. Find the sub string using start index and end index		
10	i.Given string is palindrome or not. ii.Sorting a given list of names in ascending order	2/6/2023	24
11	String handling which performs the following i. Check the capacity of the StringBuffer object. ii. Reverse the content of the this string and convert the resultant string in upper case iii. Read another string and append it .	2/6/2023	26
12	Create a class for Employee using the concept of array of Objects.	2/6/2023	28
13	Calculate the area of different shapes using the concept of method overloading.	5/6/2023	32
14	Perform Multilevel inheritance.	5/6/2023	34
15	Display function to display all the data members. Use array of objects to display details .	19/6/2023	
16	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.	19/6/2023	42
17	Write a program has class Publisher, Book, Literature and Fiction. Read the information	23/6/2023	50

	and print the details of books from either the category, using inheritance.		
18	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.	23/6/2023	55
19	Prepare bill with the given format using calculate method from interface.	3/7/2023	60
20	Prepare students marklist using inheritance and interface	3/7/2023	63
21	Create a Graphics package that has classes for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.	3/7/2023	67
22	Create an Arithmetic package that has classes for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers.	7/7/2023	70
23	Write a user defined exception class to authenticate the user name and password.	7/7/2023	72
24	Find the average of N positive integers, raising a user defined exception for each negative input.	7/7/2023	75
25	Program to find the sum of command line arguments and count the invalid integers entered through the command line.	15/7/2023	77
26	Program to create a generic stack and do the Push and Pop operations.	15/7/2023	78

27	Maintain a list of Strings using ArrayList from collection framework, perform built-in operations	15/7/2023	80
28	Program to remove all the elements from a linked list	22/7/2023	83
29	Program to find maximum of three numbers using AWT.	22/7/2023	84
30	Simple calculator using AWT components.	22/7/2023	87
31	Program to write to a file, then read from the file and display the contents on the console.	27/7/2023	90
32	Program to copy one file to another.	27/7/2023	94
33	Program that reads from a file having integers. Copy even numbers and odd numbers to separate files.	27/7/2023	96

Experiment 1

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

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

   Product(int c, String n, int p)
   {
         pcode=c;
         pname=n;
         price=p;

   }
   void display()
   {
         System.out.println("Product ID: "+pcode+"\nProduct name: "+pname+"\nPrice: 
"+price);
   }
}
class ProductDemo
{
   public static void main(String args[])
   {
         Product ob1 = new Product(101,"diarymilk",50);
         Product ob2 = new Product(102,"kitkat",30);
```

```
        Product ob3 = new Product(103,"munch",20);

        if(ob1.price<ob2.price && ob1.price<ob3.price )
        {
            System.out.println("Product with lowest price");
            ob1.display();
        }
        else if(ob2.price<ob1.price && ob2.price<ob3.price )
        {
            System.out.println("Product with lowest price");
            ob2.display();
        }
        else
        {
            System.out.println("Product with lowest price");
            ob3.display();
        }
  }
  }
```

Output

```
javac ProductDemo.java
java ProductDemo;
Product with lowest price
Product ID: 103
Product name: munch
Price: 20
```

Experiment 2

Aim

Program to demonstrate use of constructors to initialize values to member variables in a class and to display them.

Hint:- empno , empname and salary are the class members of the class employee1. From the main function we are passing the values directly to a constructor, the constructor initializes the values to member variables. The display function is used to display the stored values of the member variables.

Program

```
class Employee
{
   int eno;
   String e_name;
   int sal;

   Employee(int a,String b,int c)
   {
         eno=a;
         e_name=b;
         sal=c;
   }
   Employee()
   {
         eno=100;
         e_name="Adarsh";
         sal=10000;
   }

   void show()
   {
         System.out.println("\nEmployee NO: -"+eno+"\nEmployee Name-
"+e_name+"\nSalary- "+sal);
   }
}

class Mainclass
{
    public static void main(String args[])
    {
          Employee obj1=new Employee();
          Employee obj2=new Employee(100,"NIKHIL",35000);
          obj1.show();
          obj2.show();

    }
}
```

Output

```
javac constructor.java

javac constructor.java

Employee NO: -100

Employee Name- Adarsh

Salary- 10000



Employee NO: -100

Employee Name- NIKHIL

Salary- 35000
```

Experiment 3

Aim

Program to demonstrate use of command line arguments to initialize values to member variables in a class and to display them.

Hint:- Create a class containing Rlno, stud_name, engmark, mathsmark ,totalmark . While executing the program we have to pass arguments through command line. These values are obtained in an array which is passed as argument to main function, here it is args[ ]. The marks are converted correspondingly and then passed to constructor where values are stored to class variables. Find the total marks and later displayed using display function.

Program

```
class Student
{
   int Rlno;
   String stud_name;
   int engmark;
   int mathsmark;
   int totalmark;

   Student(int a,String b,int c,int d)
   {
         Rlno=a;
         stud_name=b;
         engmark=c;
         mathsmark=d;
   }

   void total()
   {
         totalmark=engmark+mathsmark;
         System.out.println("\nRoll no:- "+Rlno+"\nName :- "+stud_name+"\nTotal Mark
:- "+totalmark);
   }
}

class Mainclass
{
   public static void main(String args[])
   {
         int roll=Integer.parseInt(args[0]);
         int m1=Integer.parseInt(args[2]);
         int m2=Integer.parseInt(args[3]);


         Student stud1=new Student(roll,args[1],m1,m2);
         stud1.total();


   }
}
```

Output

```
javac student.java

java Mainclass 1 adarsh 40 50



Roll no:- 1

Name :- adarsh

Total Mark :- 90
```

Experiment 4

Aim

Read 2 matrices from the console and perform matrix addition.

Program

```
class matrix_op
{
   public static void main(String args[])
   {
         int arr1[][]={{1,1,1},{2,2,2},{3,3,3}};
         int arr2[][]={{1,1,2},{3,4,5},{6,1,8}};
         int sum[][]=new int[3][3];
          int i,j,k;

         System.out.println("//ADDITION ");
         for(i=0;i<3;i++){
                  for(j=0;j<3;j++)
                  {
                           sum[i][j]=arr1[i][j]+arr2[i][j];
                  }
         }
         for(i=0;i<3;i++){
                  for(j=0;j<3;j++)
                  {
                           System.out.print(sum[i][j]+" ");
                  }
                  System.out.println();
         }
           }
}
```

Output

```
11 13 15
17 19 21

23 25 27
```

Experiment 5

Aim

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

Program

```
class CmdSymmetric
{
        public static void main(String args[])
        {
         int ar[][]={{6,1,2},{5,0,9},{2,9,3}};
         int i,j,flag=1;
         int tr[][]=new int[3][3];
         System.out.println("Printing Matrix Before Transpose:");
   for(i=0;i<3;i++)
   {
         for(j=0;j<3;j++)
         {
                 System.out.print(ar[i][j]+" ");
         }
                 System.out.println();
   }

         for(i=0;i<3;i++)
         {
         for(j=0;j<3;j++)
         {
                 tr[i][j]=ar[j][i];
         }
   }
   System.out.println("Printing Matrix After Transpose:");
   for(i=0;i<3;i++)
   {
         for(j=0;j<3;j++)
         {
                 System.out.print(tr[i][j]+" ");
         }
```

```
                       System.out.println();
    }
    for (i = 0; i < 3; i++){
                      for (j = 0; j < 3; j++)
                      {
                                 if (ar[i][j] != tr[i][j])
                                 {
                                            flag = 0;
                                            break;
                                 }
                      }
    }
    if(flag==0)
    {
            System.out.println("Matrix is not symmetric");
    }
    else
    {
            System.out.println("Matrix is symmetric");
            }
   }
}
```

Output

```
javac CmdSymmetric.java
java CmdSymmetric
Printing Matrix Before Transpose:
6 1 2
5 0 9
2 9 3
Printing Matrix After Transpose:
6 5 2
1 0 9
2 9 3
Matrix is not symmetric
```

Experiment 6

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;
   cpu(double price){
         this.price=price;
   }
   public void printinfo(){
         System.out.println("cpu price is:"+price);
   }

class processor{
   int core;
   String manufacture;
   processor(int core,String manufacture){
         this.core=core;
         this.manufacture=manufacture;
   }

   public void printinfo(){
         System.out.println("cores are :"+core);
         System.out.println("manufaturer :"+manufacture);
   }
}
public static class ram{
   int memory ;
```

```
    String manufacture;
    ram(int memory,String manufacture){
    this.memory=memory;
    this.manufacture=manufacture;
    }

    public void printinfo(){
            System.out.println("ram memory"+memory);
            System.out.println("ram manufacture"+manufacture);
    }
}
    public static void main(String args []){
            cpu cpu1 =new cpu(300);
            cpu.processor processor1 = cpu1. new processor(4,"intel");
            cpu.ram ram1= new cpu.ram(8,"corsarir");
            cpu1.printinfo();
            processor1.printinfo();
            ram1.printinfo();
    }
}
```

Output

```
javac cpu.java
java cpu
CPU price is: 300.0
Cores are: 4
Manufacturer: Intel
RAM memory: 8 GB
RAM manufacturer: Corsair
```

Experiment 7

Aim

Program to Sort strings

Program

```
import java.util.Arrays;
class Sort
{
   public static void main(String args[])
   {
         String names[]={"virat","jadeja","messi","anderson","zaka"};
         int len = names.length;
         for(int i = 0; i<len-1; i++)
         {
                 for (int j = i+1; j<len; j++)
                 {
                         if(names[i].compareTo(names[j])>0)
                         {
                                 String temp = names[i];
                                 names[i] = names[j];
                                 names[j] = temp;
                         }
                 }
         }
         System.out.println(Arrays.toString(names));
   }
}
```

Output

```
javac s_sort.java
java Sort

[anderson, jadeja, messi, virat, zaka]
```

Experiment 8

Aim

Search an element in an array.

Program

```
class CmdSearch
{
   public static void main(String args[]){
          int ar[]=new int[10];
          int i,j,temp,n,l,f=0;

          int len=args.length;
          l=len-1; //size of array
          n=Integer.parseInt(args[l]); //search element
          System.out.println("Array length is: "+l);

          for(i=0;i<l;i++)
          {
                 ar[i]=Integer.parseInt(args[i]);

          }
          for(i=0;i<l;i++)
          {
                 for(j=0;j<l;j++)
                 {
                        if(ar[i]<ar[j])
                        {
                               temp=ar[i];
                               ar[i]=ar[j];
                               ar[j]=temp;
                        }

                 }
          }
          System.out.println("Given array after sorting is: ");
          for(i=0;i<l;i++)
          {
                 System.out.println(ar[i]);
```

```
            }

            System.out.println("Element to be searched is: "+n);
            for(i=0;i<l;i++)
            {
                        for(j=0;j<l;j++)
                        {
                                    if(ar[i]==n)
                                    {
                                                f=1;
                                                break;
                                    }

                        }
            }
            if(f==1)
            {
                        System.out.println("Element found ");
            }
            else{
                        System.out.println("Element not found");
            }
    }
}
```

Output

```
javac CmdSearch.java
java CmdSearch 3 5 4 2 7 4
Array length is: 5
Given array after sorting is:
2 3 4 5 7
Element to be searched is: 4
Element found
```

Experiment 9.1

Aim

perform the following operations on strings

i. Find the length of the string

ii. Character at second and fourth position

iii. Find the sub string using start index only
iv. Find the sub string using start index and end index

Program

```
class String_l
{
        public static void main(String args[])
        {
                String str = "Java Programming";

                System.out.println("The string is '"+str+"'");
                System.out.println("Length of the String is "+str.length());
                System.out.println("Character at second and fourth position is
      "+str.charAt(1)+","+str.charAt(3));
                System.out.println("The sub string using start index only is "+str.substring(5));
                System.out.println("The sub string using start index only is "+str.substring(0,4));
        }
}
```

Output

```
javac string_l.java
java String_l
The string is 'Java Programming'
Length of the String is 16
Character at second and fourth position is a,a
The sub string using start index only is Programming
The sub string using start index only is Java
```

Experiment 9.2

Aim

Perform the following operations on strings

i. compare two strings lexicographically.

Program

```
 class lexicograph
 {
       public static void main(String args[])
       {
         int ar[] = new int[10];
         String s1 = args[0],s2=args[1];
         int largest = s1.toLowerCase().compareTo(s2.toLowerCase());
         System.out.println(largest);
          if(largest<0)
             {
                System.out.println(s1+ " is greater than " +s2);
              }
          else if(largest==0)
           {
              System.out.println(s1+" is equal to " +s2);
           }
          else
          {
              System.out.println(s1+" is less than"+s2);
          }
      }
}
```

Output

```
javac lexicograph.java
java lexicograph haritha bhagya
6
haritha is greater than bhagya
```

Experiment 9.3

Aim

Perform the following operation on strings

i. compare two strings lexicographically, ignoring case differences.

Program

```
class lexico

{

    public static void main(String args[])

    {

            int ar[] = new int[10];

                       String s1 = args[0],s2=args[1];

                       int largest =s1.compareToIgnoreCase(s2);

                       System.out.println(largest);

                       if(largest>0)

                       {

                                  System.out.println(s1+ " is greater than " +s2);

           }

           else if(largest==0)

           {

                       System.out.println(s1+" is equal to " +s2);

           }

           else
    {
            System.out.println(s1+" is less than"+s2);
        }
    }
}
```

Output

```
javac lexico.java

java lexico soman ADARSH

18

soman is greater than ADARSH
```

Experiment 9.4

Aim

Perform the following operation on strings

i. concatenate a given string to the end of another string.

Program

```
class ConString{
    public static void main(String[] args) {
        String s1="Hello";
        String s2="World";
        String s3=s1.concat(s2);
        System.out.println(s3);

        String s4="Good"+"Morning";
         System.out.println(s4);
    }
}
```

Output

```
HelloWorld

GoodMorning
```

Experiment 9.5

Aim

Perform the following operation on strings

i. Replace a specified character with another character.

Program

```
class ReplaceString{

    public static void main(String[] args) {

        String s1="Today is monday";

        System.out.println("String is : "+s1);

        String s2=s1.replace("is","=");

        System.out.println(s2);

    }

}
```

Output

```
String is : Today is monday

Today = monday
```

Experiment 9.6

Aim

Perform the following operation on strings

i. check whether a given string starts with another string.

Program

```
class string
        {
           public static void main(String args[])
           {
                  String str = args[0];
                 String toCheckStr = "hello";
                 if(str.startsWith(toCheckStr))
                  {
                                  System.out.println("The string does starts with "+toCheckStr);
                  }
                 else {
                                  System.out.println("The string does not starts with "+toCheckStr);
                 }
           }
        }
```

Output

```
javac start_with.java

java string goodmorning

The string does not starts with hello
```

Experiment 9.7

Aim

Perform the following operation on strings

i. convert all characters in a string to lowercase.

ii. convert all characters in a string to uppercase.

Program

```
class StartString{

   public static void main(String[] args) {

      String s1="Good Morning";

      System.out.println(s1);



      System.out.println("To upper case: "+s1.toUpperCase());

      System.out.println("\nTo lower case: "+s1.toLowerCase());

     }

}
```

Output

```
Good Morning

To upper case: GOOD MORNING

To lower case: good morning
```

Experiment 10.1

Aim

Write a java program to

i. check whether a given string is palindrome or not.

Program

```
class Palindrome
{
   public static void main(String args[])
   {
         String str=args[0];
         String rev="";
         int len=str.length();
         for(int i=len-1; i>=0;i--)
         {
          rev=rev+str.charAt(i);
         }
         if(rev.toLowerCase().equals(str.toLowerCase()))
         {
                 System.out.println("The string is Palindrome");
         }
         else{
                 System.out.println("The string is not Palindrome");
         }
   }
}
```

Output

```
javac palindrome_s.java
java Palindrome malayalam
The string is Palindrome

java Palindrome car
The string is not Palindrome
```

Experiment 10.2

Aim

Write a java program to

i. sorting a given list of names in ascending order

Program

```
import java.util.Arrays;
class Sort
{
    public static void main(String args[])
    {
            String names[]={"virat","jadeja","messi","anderson","zaka"};
            int len = names.length;
            for(int i = 0; i<len-1; i++)
            {
                    for (int j = i+1; j<len; j++)
                    {
                            if(names[i].compareTo(names[j])>0)
                            {
                                    String temp = names[i];
                                    names[i] = names[j];
                                    names[j] = temp;
                            }
                    }
            }
            System.out.println(Arrays.toString(names));
    }
}
```

Output

```
javac s_sort.java
java Sort
[anderson, jadeja, messi, virat, zaka]
```

Experiment 11

Aim

Write a program in java for string handling which performs the following

i. Check the capacity of the StringBuffer object.

ii. Reverse the content of the this string and convert the resultant string in upper case

iii. Read another string and append it to the resultant string of abov**e.**

Program

```
public class StringBufferop {

        public static void main(String[] args)

        {

                StringBuffer sb = new StringBuffer("Java Programming");

                System.out.println("(i)The capacity of string buffer object is
                "+sb.capacity());

                String str=sb.reverse().toString();

                   System.out.println("(ii)Reversed string buffer object is
"+str.toUpperCase());

             System.out.println("(iii)String after append "+sb.append(" Language"));

        }

}
```

Output

```
Javac StringBufferop.java

Java  StringBufferop

(i)The capacity of string buffer object is 32

(ii)Reversed string buffer object is GNIMMARGORP AVAJ

(iii)String after append gnimmargorP avaJ Language
```

Experiment 12

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

```
import java.util.*;

class Employee
{
        int eNo, eSalary;
        String eName;

        Employee(String a, int b, int c) {
        eName = a;
        eNo = b;
        eSalary = c;
        }

        int return_id()
        {
        return eNo;
        }
}

class Company
{
        public static void main(String args[])
        {
        Scanner s = new Scanner(System.in);
        Employee E[] = new Employee[5];
```

```
int n, id, salary;
String name1;
System.out.println("Enter number of employees");
n = s.nextInt();
s.nextLine();
for (int i = 0; i < n; i++) {
System.out.println("Enter the employee details");
System.out.println("Enter the employee id");
id = s.nextInt();
s.nextLine();
System.out.println("Enter the employee Name");
name1 = s.nextLine();
System.out.println("Enter the employee Salary");
salary = s.nextInt();
s.nextLine();

E[i] = new Employee(name1, id, salary);
}
System.out.println("Enter Id of employees to be searched");
int ele = s.nextInt();
int h, f = 0;
for (int i = 0; i < n; i++) {
h = E[i].return_id();
if (ele == h) {
        f = f + 1;
}
}
if (f > 0) {
System.out.println("The Employee having the id " + ele + " is present");
} else {
System.out.println("The Employee having the id " + ele + " is not present");
}
```

```
        }
}
```

Output

```
javac emp_search.java

java Company

Enter number of employees
3
Enter the employee details
Enter the employee id
12
Enter the employee Name
ram
Enter the employee Salary
15000
Enter the employee details
Enter the employee id
122
Enter the employee Name
ravi
Enter the employee Salary
20000
Enter the employee details
Enter the employee id
111
Enter the employee Name
mohan
Enter the employee Salary
15000
```

```
Enter Id of employees to be searched
111
The Employee having the id 111 is present
```

CO 3

Experiment 13

Aim

Write a java program to calculate the area of different shapes namely circle, rectangle and triangle using the concept of method overloading.

Program

```
class Shapes

{

   int x,y;

   double z;

   void cal(int l){

         x=l;

         System.out.println("Area of Square: "+x*x);

   }

   void cal(int l,int b){a

         x=l;

         y=b;

         System.out.println("Area of Triangle: "+0.5*x*y);

   }

   void cal(double r)

   {
        z=r;

        System.out.println("Area of Circle: "+3.14*z*z);

    }

}

class ShapesDemo{

    public static void main(String args[]){

        Shapes s = new Shapes();

        s.cal(5);

        s.cal(2,4);

        s.cal(2.5);

    }

}
```

Output

```
javac ShapesDemo.java

java ShapesDemo

Area of Square: 25

Area of Triangle: 4.0

Area of Circle: 19.625
```

Experiment 14

Aim

Perform Multilevel inheritance.

Hint:- The base class ‘stud_details’ is created for assigning the Rol.no and name of the student and display it. The derived class “Marks” is created for inputting 3 subjects marks and display it. Again a derived class is created as “Total” for calculate the total mark of 3 subjects and display it. Inputs are given through the keyboard.

Program

```
class Stud_details

{

   int r;

   String n;

   void std(int roll,String name)

   {

         r=roll;

         n=name;

         System.out.println("Roll no: "+r+" Name: "+n);

   }

}

class Marks extends Stud_details

{

   int m1,m2,m3;
```

```
    void mark(int mark1,int mark2,int mark3)

    {

            m1=mark1;

            m2=mark2;

            m3=mark3;

            System.out.println("Mark1= "+m1+" Mark2= "+m2+" Mark3= "+m3);

    }

}

class Total extends Marks

{

    int t;

    void getTotal()

    {

            t=m1+m2+m3;

            System.out.println("Total marks = "+t);

    }

}

class StdDemo

{

    public static void main(String args[])
    {

            Total s= new Total();

            s.std(101,"Nidhi");

            s.mark(70,80,85);

            s.getTotal();

    }

}
```

Output

```
javac StdDemo.java

java StdDemo

Roll no: 101 Name: Nidhi

Mark1= 70 Mark2= 80 Mark3= 85

Total marks = 235
```

Experiment 15

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.*;

class Employee

{

   int empid,salary;

   String name,address;


   Employee(int e,String n,int s,String a)

   {

         empid=e;

         name=n;

         salary=s;

         address=a;

   }

}

class Teacher extends Employee

{

   String department, subject;


   Teacher(int e,String n,int s,String a,String d,String sub)
```

```
    {
        super(e,n,s,a);
        department=d;
        subject=sub;
    }
    void display()
    {
        System.out.println("ID: "+empid);
        System.out.println("Name: "+name);
        System.out.println("Salary: "+salary);
        System.out.println("Address: "+address);
        System.out.println("Department: "+department);
        System.out.println("Subject: "+subject);
    }

}
class arrayObject
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        Teacher T[] = new Teacher[5];
        int num,id,sal;
        String empname,empadd,empdept,empsub;
        System.out.println("Enter number of employees: ");
```

```
num=s.nextInt();
for(int i=0;i<num;i++)
{
        System.out.println("Enter employee ID: ");
        id=s.nextInt();
        s.nextLine();

        System.out.println("Enter employee name: ");
        empname=s.nextLine();

        System.out.println("Enter employee salary: ");
        sal=s.nextInt();
        s.nextLine();

        System.out.println("Enter employee address: ");
        empadd=s.nextLine();

        System.out.println("Enter employee department: ");
        empdept=s.nextLine();

        System.out.println("Enter employee subject: ");
        empsub=s.nextLine();

        T[i] = new Teacher(id,empname,sal,empadd,empdept,empsub);
```

```
            }
            for(int i=0;i<num;i++)
            {
                    T[i].display();
            }

    }
}
```

Output

```
javac arrayObject.java
java arrayObject
Enter number of employees:
1
Enter employee ID:
101
Enter employee name:
bhagya
Enter employee salary:
10000
Enter employee address:
kochi
Enter employee department:
maths
Enter employee subject:
```

```
maths
ID: 101
Name: bhagya
Salary: 10000
Address: kochi
Department: maths
Subject: maths
```

Experiment 16

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.Scanner;

class Person {

    protected String name;

    protected String gender;

    protected String address;

    protected int age;

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

    name = name;

    gender = gender;

    address = address;

    age = age;

    }

    public void display() {

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

```
        System.out.println("Gender: " + gender);

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

        System.out.println("Age: " + age);

        }

}

class Employee extends Person {

        protected int empid;

        protected String companyName;

        protected String qualification;

        protected double salary;

        public Employee(String name, String gender, String address, int age, int empid, String
companyName, String qualification, double salary) {

        super(name, gender, address, age);

        empid = empid;

        companyName = companyName;

        qualification = qualification;

        salary = salary;

        }

        public void display() {

        super.display();

        System.out.println("Employee ID: " + empid);
        System.out.println("Company Name: " + companyName);
        System.out.println("Qualification: " + qualification);
        System.out.println("Salary: " + salary);
        }
}
class Teacher extends Employee {
        protected String subject;
        protected String department;
        protected int teacherId;
        public Teacher(String name, String gender, String address, int age, int empid, String companyName, String qualification, double salary, String subject, String department, int teacherId) {
        super(name, gender, address, age, empid, companyName, qualification, salary);
        subject = subject;
        department = department;
        teacherId = teacherId;
        }
        public void display() {
        super.display();
        System.out.println("Subject: " + subject);
        System.out.println("Department: " + department);
```

```
    System.out.println("Teacher ID: " + teacherId);
    }
}
public class TeacherManagementSystem {
    public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter the number of teachers (N): ");
    int n = scanner.nextInt();
    Teacher[] teachers = new Teacher[n];
    for (int i = 0; i < n; i++) {
    System.out.println("Enter details for Teacher #" + (i + 1));
    scanner.nextLine();
    System.out.print("Name: ");
    String name = scanner.nextLine();
    System.out.print("Gender: ");
    String gender = scanner.nextLine();
    System.out.print("Address: ");
    String address = scanner.nextLine();
    System.out.print("Age: ");
    int age = scanner.nextInt();
```

```
System.out.print("Employee ID: ");

int empid = scanner.nextInt();

scanner.nextLine();

System.out.print("Company Name: ");

String companyName = scanner.nextLine();

System.out.print("Qualification: ");

String qualification = scanner.nextLine();

System.out.print("Salary: ");

double salary = scanner.nextDouble();

scanner.nextLine();

System.out.print("Subject: ");

String subject = scanner.nextLine();

System.out.print("Department: ");

String department = scanner.nextLine();

System.out.print("Teacher ID: ");

int teacherId = scanner.nextInt();

teachers[i] = new Teacher(name, gender, address, age, empid, companyName, qualification, salary, subject, department, teacherId);

}

System.out.println("\nDetails of Teachers:");

for (int i = 0; i < n; i++) {
    System.out.println("\nTeacher #" + (i + 1));

    teachers[i].display();

    }

    }

}
```

Output

```
javac TeacherManagementSystem.java

java TeacherManagementSystem

Enter the number of teachers (N): 2

Enter details for Teacher #1

Name: mike

Gender: male

Address: Ernakulam

Age: 49

Employee ID: 1

Company Name: muthoot

Qualification: bachelors degree in biology

Salary: 40000

Subject: biology

Department: science
```

```
Teacher ID: 1
Enter details for Teacher #2
Name: lisa
Gender: female
Address: kochi
Age: 34
Employee ID: 2
Company Name: central high school
Qualification: bachelors degree in mathematics
Salary: 34000
Subject: algebra
Department: mathematics
Teacher ID:
Details of Teachers:
Teacher #1
Name: mike
Gender: male
Address: eranakulam
Age: 49
Employee ID: 1
```

Company Name: muthoot

Qualification: bachelors degree in biology

Salary: 40000

Subject: biology

Department: science

Teacher ID: 1

Teacher #2

Name: lisa

Gender: female

Address: kochi

Age: 34

Employee ID: 2

Company Name: central high school

Qualification: bachelors degree in mathematics

Salary: 34000

Subject: algebra

Department: mathematics

Teacher ID: 2

Experiment 17

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.ArrayList;

import java.util.List;

import java.util.Scanner;



class Publisher {

        private String name;

        public Publisher(String name) {

        name = name; // Removed "this" keyword

        }

        public String getName() {

        return name;

        }

}

class Book {

        private String title;

        private double price;
        private Publisher publisher;
        public Book(String title, double price, Publisher publisher) {
        title = title; // Removed "this" keyword
        price = price; // Removed "this" keyword
        publisher = publisher; // Removed "this" keyword
        }
        public String getTitle() {
        return title;
        }
        public double getPrice() {
        return price;
        }
        public Publisher getPublisher() {
        return publisher;
        }
        public void display() {
        System.out.println("Title: " + title);
        System.out.println("Price: $" + price);
        System.out.println("Publisher: " + publisher.getName());
        }
```

```
}
class Literature extends Book {
    public Literature(String title, double price, Publisher publisher) {
    super(title, price, publisher);
    }
}
class Fiction extends Book {
    public Fiction(String title, double price, Publisher publisher) {
    super(title, price, publisher);
    }
}
public class Main {
    public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    List<Book> books = new ArrayList<>();
    Publisher publisher1 = new Publisher("Publisher A");
    Publisher publisher2 = new Publisher("Publisher B");
    books.add(new Literature("Literature Book 1", 20.99, publisher1));
    books.add(new Literature("Literature Book 2", 18.50, publisher1));
    books.add(new Fiction("Fiction Book 1", 15.75, publisher2));
```

```
        books.add(new Fiction("Fiction Book 2", 12.99, publisher2));

        System.out.println("Enter the category (Literature or Fiction): ");

        String category = scanner.nextLine();

        System.out.println("Books in the " + category + " category:");

        for (Book book : books) {

        if ((category.equalsIgnoreCase("Literature") && book instanceof Literature)

                || (category.equalsIgnoreCase("Fiction") && book instanceof Fiction)) {

                book.display();

                System.out.println();

        }

        }

        scanner.close();

        }

}
```

Output

```
javac book_details.java

java book_details


Enter the category (Literature or Fiction):
```

Fiction

Books in the Fiction category:

Title: Fiction Book 1

Price: $15.75

Publisher: Publisher B

Title: Fiction Book 2

Price: $12.99

Publisher: Publisher B

Experiment 18

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

```
import java.util.Scanner;


interface Shape {
    double area();
    double perimeter();
}


class Circle implements Shape {
    private double radius;


    public Circle(double radius) {
        this.radius = radius;
    }



    public double area() {
        return Math.PI * radius * radius;
    }
    public double perimeter() {
        return 2 * Math.PI * radius;
    }
}

class Rectangle implements Shape {
    private double length;
    private double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }


    public double area() {
        return length * width;
    }


    public double perimeter() {
        return 2 * (length + width);
    }
}

public class Area_Perimeter {
```

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

    while (true) {
        System.out.println("Menu:");
        System.out.println("1. Calculate Circle");
        System.out.println("2. Calculate Rectangle");
        System.out.println("3. Exit");
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();

        switch (choice) {
            case 1:
                System.out.print("Enter the radius of the circle: ");
                double radius = scanner.nextDouble();
                Circle circle = new Circle(radius);
                System.out.println("Area of Circle: " + circle.area());
                System.out.println("Perimeter of Circle: " + circle.perimeter());
                break;

            case 2:
                System.out.print("Enter the length of the rectangle: ");
                double length = scanner.nextDouble();
                System.out.print("Enter the width of the rectangle: ");
                double width = scanner.nextDouble();
                Rectangle rectangle = new Rectangle(length, width);
                System.out.println("Area of Rectangle: " + rectangle.area());
```

```
                    System.out.println("Perimeter of Rectangle: " + rectangle.perimeter());
                    break;
                case 3:
                    System.out.println("Exiting the program.");
                    scanner.close();
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please try again.");
                    break;
            }
        }
    }
}
```

Output

```
javac  Area_Perimeter.java
java  Area_Perimeter
```

```
Menu:
1. Calculate Circle
2. Calculate Rectangle
3. Exit
Enter your choice: 1
Enter the radius of the circle: 10
Area of Circle: 314.1592653589793
Perimeter of Circle: 62.83185307179586
```

```
Menu:
1. Calculate Circle
2. Calculate Rectangle
3. Exit
Enter your choice: 2
Enter the length of the rectangle: 10
Enter the width of the rectangle: 20
Area of Rectangle: 200.0
Perimeter of Rectangle: 60.0
Menu:
1. Calculate Circle
2. Calculate Rectangle
3. Exit
Enter your choice: 3
Exiting the program.
```

Experiment 19

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

```
class Product {
    private int productId;
    private String name;
    private int quantity;
    private double unitPrice;

    public Product(int id, String productName, int productQuantity, double productUnitPrice) {
        productId = id;
        name = productName;
        quantity = productQuantity;
        unitPrice = productUnitPrice;
    }

    public double calculateTotal() {
        return quantity * unitPrice;
    }



    public void displayProduct() {
        System.out.printf("%-10d%-10s%-10d%-15.2f%-10.2f%n", productId, name, quantity,
unitPrice, calculateTotal());
    }
}

public class BillWithoutInterface {
    public static void main(String[] args) {
        // Create instances of products
        Product product1 = new Product(101, "A", 2, 25.0);
        Product product2 = new Product(102, "B", 1, 100.0);

        // Calculate the total net amount for each product
        double total1 = product1.calculateTotal();
        double total2 = product2.calculateTotal();

        // Calculate the net amount for the entire bill
        double netAmount = total1 + total2;

        // Display the bill format
        System.out.println("Order No.   Date :");
        System.out.println("Product Id   Name    Quantity    Unit Price     Total");
        product1.displayProduct();
        product2.displayProduct();
        System.out.printf("%-50s%-10.2f%n", "Net. Amount", netAmount);
    }
}
```

Output

```
Order No.   Date :
Product Id   Name    Quantity    Unit Price     Total
   101            A          2              25.00          50.00
   102            B           1          100.00        100.00
              Net. Amount                              150.00
```

Experiment 20

Aim Prepare the students mark list using inheritance and interface concepts.

Program

```
import java.util.Scanner;
class stud_details {
   private int Rol_no;
   private String student_name;
   public int maths_mark;
   public int chemistry_mark;
   public int physics_mark;

   stud_details(String name, int num, int m1, int m2, int m3) {
      student_name = name;
      Rol_no = num;
      maths_mark = m1;
      chemistry_mark = m2;
      physics_mark = m3;
   }
   void display() {
      System.out.println("Name of student is " + student_name);
      System.out.println("Roll number of student is " + Rol_no);
      System.out.println("Maths marks is " + maths_mark);
      System.out.println("Chemistry marks is " + chemistry_mark);
      System.out.println("Physics marks is " + physics_mark);
   }
```

```
}
interface Marks {
    public void calc();
}
class student extends stud_details implements Marks {
    student(String name, int num, int m1, int m2, int m3) {
        super(name, num, m1, m2, m3);
    }
    public void calc() {
        int total_mark = maths_mark + chemistry_mark + physics_mark;
        int percentage = (total_mark * 100) / 300;
        System.out.println("Total mark is " + total_mark);
        System.out.println("Percentage is " + percentage);
        if (percentage >= 80) {
            System.out.println("Your grade is A");
        } else if (percentage >= 70 && percentage < 80) {
            System.out.println("Your grade is B");
        } else if (percentage >= 60 && percentage < 70) {
            System.out.println("Your grade is C");
        } else if (percentage >= 50 && percentage < 60) {
            System.out.println("Your grade is D");
        } else {
            System.out.println("FAIL");
        }
    }
    void display() {
        super.display();
    }
}
class MarkList {
    public static void main(String arg[]) {
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter name of student ");
        String name = obj.nextLine();
        System.out.println("Enter roll number ");
        int num = obj.nextInt();
        System.out.println("Enter maths mark");
        int m1 = obj.nextInt();
        System.out.println("Enter physics mark");
        int m2 = obj.nextInt();
        System.out.println("Enter chemistry mark");
        int m3 = obj.nextInt();
        student stud = new student(name, num, m1, m2, m3);
        stud.display();
        stud.calc();
    }
}
```

Output

```
javac MarkList.java
java MarkList

Enter name of student
Arun
Enter roll number
```

```
14
Enter maths mark
88
Enter physics mark
79
Enter chemistry mark
90
Name of student is Arun
Roll number of student is 14
Maths marks is 88
Chemistry marks is 79
Physics marks is 90
Total mark is 257
Percentage is 85
Your grade is A
```

CO 4

Experiment 21

Aim

Create a Graphics package that has classes for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

Program

Graphic Package

rectangle.java

```
package Graphic;
public class rectangle {
    private double length;
    private double width;
    public rectangle(double len, double wid) {
        length = len;
        width = wid;
    }
    public double calculateArea() {
        return length * width;
    }
}
```

Triangle.java

```
package Graphic;
public class Triangle {
    private double base;
    private double height;
    public Triangle(double b, double h) {
        base = b;
        height = h;
    }
    public double calculateArea() {
        return 0.5 * base * height;
    }
}
```

Square.java

```
package Graphic;
public class Square {
   private double side;
   public Square(double s) {
      side = s;
   }
   public double calculateArea() {
      return side * side;
   }
}
```

Circle.java

```
package Graphic;
public class Circle {
   private double radius;
   public Circle(double r) {
      radius = r;
   }
   public double calculateArea() {
      return Math.PI * radius * radius;
   }
}
```

TestShape.java

```
import Graphic.rectangle;
import Graphic.Triangle;
import Graphic.Square;
import Graphic.Circle;
public class TestShapes {
   public static void main(String[] args) {

      rectangle rectangle = new rectangle(5, 3);
      Triangle triangle = new Triangle(4, 6);
      Square square = new Square(4);
      Circle circle = new Circle(3);
      System.out.println("Area of Rectangle: " + rectangle.calculateArea());
      System.out.println("Area of Triangle: " + triangle.calculateArea());
      System.out.println("Area of Square: " + square.calculateArea());
      System.out.println("Area of Circle: " + circle.calculateArea());
```

```
    }
}
```

Output

```
Javac TestShapes.java
Java TestShapes

Area of Rectangle: 15.0
Area of Triangle: 12.0
Area of Square: 16.0
Area of Circle: 28.259999999999998
```

Experiment 22

Aim

Create an Arithmetic package that has classes for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers.

Program

Arithmetic package

Add.java:-

```
package arithmatic;
public class Add
{
   public int sum(int x,int y)
   {
    return x+y;
   }
}
```

div.java

```
package arithmatic;
public class Div
{
   public double Quotient(int x,int y)
   {
    return x/y;
   }
}
```

mult.java

```
package arithmatic;
```

```
public class Mult
{
   public int Product(int x,int y)
   {
    return x*y;
   }
}
```

Subtract.java

```
package arithmatic;
public class Subtract
{
   public int diff(int x,int y)
   {
    return x-y;
   }
}
```

operations.java

```
import arithmatic.*;
class Operations
{
   public static void main(String[] args) {
         Add ob1 = new Add();
         Subtract ob2 = new Subtract();
         Mult ob3 = new Mult();
         Div ob3 = new Div();
         int s=ob1.sum();

   }
}
```

Experiment 23

Aim

Write a user defined exception class to authenticate the user name and password.

Program

```
import java.util.Scanner;

class UsernameException extends Exception

{

   public UsernameException(String msg) {

     super(msg);

            }

}

class PasswordException extends Exception

{

   public PasswordException(String msg) {

     super(msg);

            }

}

class DemoExcp

{

   public static void main(String args[])

   {
```

```
String name="nidhi";
 String pass="nidhips";
 Scanner s= new Scanner(System.in);
System.out.print("Enter username :: ");
String user=s.nextLine();
System.out.print("Enter password :: ");
String pwd=s.nextLine();
try
{
        if(!user.equals(name))
                throw new UsernameException("Username incorrect");
        else if(!pwd.equals(pass))
                throw new PasswordException("Password incorrect");
         else
                        System.out.println("Login Successful !!!");
}
catch(UsernameException u)
{
        u.printStackTrace();
}
```

```
            catch(PasswordException p)

            {

                        p.printStackTrace();

            }

    }

}
```

Output

```
java DemoException
Enter username :: nidhi
Enter password :: nidhips
Login Successful !!!
```

Experiment 24

Aim

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

Program

```
import java.util.Scanner;

class NegativeInputException extends Exception {
    public NegativeInputException() {
    super("Negative input is not allowed.");
    }
}

class Average{
    public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the value of N: ");
    int N = scanner.nextInt();

    int sum = 0;
    int count = 0;

    for (int i = 1; i <= N; i++) {
          try {
                    System.out.print("Enter a integer: ");
                    int num = scanner.nextInt();

                    if (num < 0) {
                    throw new NegativeInputException();
                    }
```

```
            sum += num;
            count++;
        } catch (NegativeInputException e) {
            System.out.println(e.getMessage());
            i--;
        }
    }
        double average=sum/N;
        System.out.println("The Average of the numbers is: "+average);
    }
    }
```

Output

```
javac neg.java
java Average

Enter the value of N: 3
Enter a integer: 1
Enter a integer: -3
Negative input is not allowed.
Enter a integer: 2
Enter a integer: -5
Negative input is not allowed.
Enter a integer: 3
The Average of the numbers is: 2.0
```

Experiment 25

Aim

Program to find the sum of command line arguments and count the invalid integers   entered through the command line.

Program

```
class cmdSum{
   public static void main(String arg[]){
         int sum=0;
         int invalid=0;
         int num=0;
         for(int i=0;i<arg.length;i++){
                  try {
                           num=Integer.parseInt(arg[i]);
                           sum=num+sum;
                  }
                  catch(NumberFormatException e){
                           invalid=invalid+1;
                  }
         }
         System.out.println("sum = " + sum);
         System.out.println("invalid integers = " + invalid);
   }
}
```

Output

```
java cmdSum 2 4 6 7 s b 8
sum = 27
invalid integers = 2
```

EXPERIMENT 26

Aim

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

Program

```
import java.util.*;

import java.io.*;

public class StackDemo {

   public static void main(String args[])

   {

         // Creating an empty Stack

         Stack<Integer> stk = new Stack<Integer>();

         // Use add() method to add elements

         stk.push(10);

         stk.push(15);

         stk.push(30);

         stk.push(20);

         stk.push(5);

         // Displaying the Stack

         System.out.println("Initial Stack: " + stk);

         // Fetching the element at the head of the Stack

         System.out.println("The element at the top of the stack is: " +
stk.peek());

         // Removing elements using pop() method

         System.out.println("Popped element: " + stk.pop());

         System.out.println("Popped element: " + stk.pop());
```

```
            // Displaying the Stack after pop operation
            System.out.println("Stack after pop operation: " + stk);
            System.out.println("The element at the top of the stack is: " +
stk.peek());
    }
}
```

Output

```
java StackDemo
Initial Stack: [10, 15, 30, 20, 5]
The element at the top of the stack is: 5
Popped element: 5
Popped element: 20
Stack after pop operation: [10, 15, 30]
The element at the top of the stack is: 30
```

EXPERIMENT 27

Aim

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

Program

```
import java.util.ArrayList;

import java.util.Collections;

import java.util.Iterator;

import java.util.List;

public class ArrayListDemo {

    public static void main(String[] args) {

        // Create an ArrayList of Strings

        List<String> stringList = new ArrayList<>();

        // Add elements to the ArrayList

        stringList.add("Apple");

        stringList.add("Banana");

        stringList.add("Cherry");

        stringList.add("Date");

        // Display the ArrayList

        System.out.println("ArrayList: " + stringList);

        // Access elements by index

        System.out.println("Element at index 2: " + stringList.get(2));

        // Modify an element

        stringList.set(1, "Blueberry");

        System.out.println("Modified ArrayList: " + stringList);
```

```
// Check if an element exists in the ArrayList
boolean containsCherry = stringList.contains("Cherry");
System.out.println("Contains 'Cherry'? " + containsCherry);
// Find the index of an element
int indexOfDate = stringList.indexOf("Date");
System.out.println("Index of 'Date': " + indexOfDate);
// Remove an element by value
boolean removed = stringList.remove("Banana");
System.out.println("Removed 'Banana'? " + removed);
System.out.println("ArrayList after removal: " + stringList);
// Iterate through the ArrayList
System.out.println("Iterating through the ArrayList:");
for (String fruit : stringList) {
    System.out.println(fruit);
}
// Sort the ArrayList
Collections.sort(stringList);
System.out.println("Sorted ArrayList: " + stringList);
// Check if the ArrayList is empty
boolean isEmpty = stringList.isEmpty();
System.out.println("Is the ArrayList empty? " + isEmpty);
// Get the size of the ArrayList
int size = stringList.size();
System.out.println("Size of the ArrayList: " + size);
```

```
        // Clear the ArrayList
        stringList.clear();
        System.out.println("Cleared ArrayList: " + stringList);
    }
}
```

Output

```
javac ArrayListDemo.java
java ArrayListDemo

Index of 'Date': 3
Removed 'Banana'? false
ArrayList after removal: [Apple, Blueberry, Cherry, Date]
Iterating through the ArrayList:
Apple
Blueberry
Cherry
Date
Sorted ArrayList: [Apple, Blueberry, Cherry, Date]
Is the ArrayList empty? false
Size of the ArrayList: 4
Cleared ArrayList: []
```

EXPERIMENT 28

Aim

Program to remove all the elements from a linked list

Program

```
import java.util.LinkedList;
public class RemoveAll {
    public static void main(String[] args) {
        // Create a LinkedList
        LinkedList<String> linkedList = new LinkedList<>();
        // Add elements to the LinkedList
        linkedList.add("Apple");
        linkedList.add("Banana");
        linkedList.add("Cherry");
        linkedList.add("Date");
        // Display the initial LinkedList
        System.out.println("Initial LinkedList: " + linkedList);
        // Remove all elements from the LinkedList
        linkedList.clear();
        // Display the LinkedList after removal
        System.out.println("LinkedList after removing all elements: " + linkedList);
    }
}
```

Output

```
javac RemoveAll.java
java RemoveAll
Initial LinkedList: [Apple, Banana, Cherry, Date]
LinkedList after removing all elements: []
```

CO5

EXPERIMENT 29

Aim

Program to find maximum of three numbers using AWT.

Program

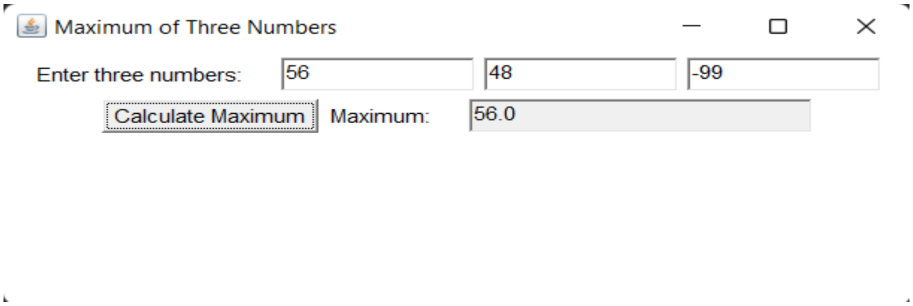
```
import java.awt.*;
import java.awt.event.*;

public class MaxNumber extends Frame {
    private TextField num1Field, num2Field, num3Field, resultField;
    private Button calculateButton;
    public MaxNumber()
        setTitle("Maximum of Three Numbers");
        setSize(300, 200);
        setLayout(new FlowLayout());
        num1Field = new TextField(10);
        num2Field = new TextField(10);
        num3Field = new TextField(10)
        resultField = new TextField(20);
        resultField.setEditable(false);
        calculateButton = new Button("Calculate Maximum");
        add(new Label("Enter three numbers: "));
        add(num1Field);
        add(num2Field);
        add(num3Field);
        add(calculateButton);
        add(new Label("Maximum: "));
        add(resultField);
        calculateButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                calculateMaximum();
            }
        });
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }
    private void calculateMaximum() {
        try {

            double num1 = Double.parseDouble(num1Field.getText());
            double num2 = Double.parseDouble(num2Field.getText());
            double num3 = Double.parseDouble(num3Field.getText());
            double maximum = Math.max(Math.max(num1, num2), num3);
            resultField.setText(Double.toString(maximum));
        } catch (NumberFormatException ex) {
            // Handle invalid input (non-numeric)
            resultField.setText("Invalid input. Enter numbers.");
        }
    }
    public static void main(String[] args) {
        MaxNumber maxApp = new MaxNumber();
        maxApp.setVisible(true);
    }
}
```

Output

```
javac MaxNumber.java
java MaxNumber
```



Maximum of Three Numbers

Enter three numbers: 56 48 -99

Maximum: 56.0

EXPERIMENT 30

Aim

Implement a simple calculator using AWT components.

Program

```
import java.awt.*;
import java.awt.event.*;
class calculator implements ActionListener{
        Frame f=new Frame();
        Label l1=new Label("First Number");
        Label l2=new Label("Second Number");
        Label l3=new Label("Result");
        TextField t1=new TextField();
        TextField t2=new TextField();
        TextField t3=new TextField();
        Button b1=new Button("Add");
        Button b2=new Button("Sub");
        Button b3=new Button("Mul");
        Button b4=new Button("Div");
        Button b5=new Button("Cancel");
        calculator(){
        l1.setBounds(50,100,100,20);
        l2.setBounds(50,140,100,20);
        l3.setBounds(50,180,100,20);
        t1.setBounds(200,100,100,20);
        t2.setBounds(200,140,100,20);
        t3.setBounds(200,180,100,20);
        b1.setBounds(50,250,50,20);
        b2.setBounds(110,250,50,20);
        b3.setBounds(170,258,50,20);
        b4.setBounds(230,250,50,20);
        b5.setBounds(290,250,50,20);
        f.add(l1);
        f.add(l2);
        f.add(l3);
        f.add(t1);
```

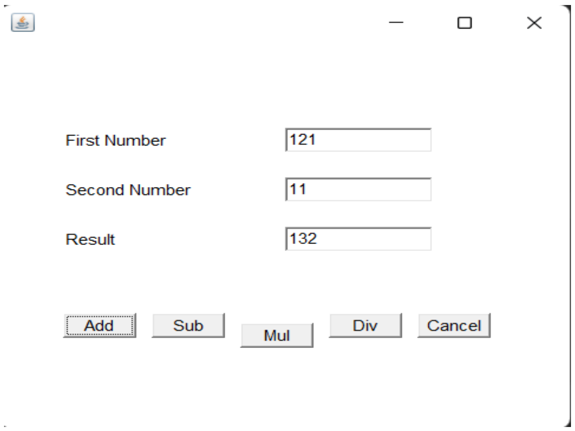
```
    f.add(t2);
    f.add(t3);
    f.add(b1);
    f.add(b2);
    f.add(b3);
    f.add(b4);
    f.add(b5)
    b1.addActionListener(this);
    b2.addActionListener(this);
    b3.addActionListener(this);
    b4.addActionListener(this);
    b5.addActionListener(this);
    f.setLayout(null);
    f.setVisible(true);
    f.setSize(400,350);
    f.setLocation(500,200);
    }
public void actionPerformed(ActionEvent e) {
    int n1 =Integer.parseInt(t1.getText());
    int n2 =Integer.parseInt(t2.getText());
    if (e.getSource() == b1) {
        t3.setText(String.valueOf(n1 + n2));
    }
    if (e.getSource() == b2) {
        t3.setText(String.valueOf(n1 - n2));
    }
    if (e.getSource() == b3) {
        t3.setText(String.valueOf(n1 * n2));
    }
    if (e.getSource() == b4) {
        t3.setText(String.valueOf(n1 / n2));
    }
    if (e.getSource() == b5) {
        System.exit(0);
    }
}
public static void main(String[] args)  {
new calculator();
```

```
    }
}
```

Output

```
javac calculator.java
java calculator
```

Add

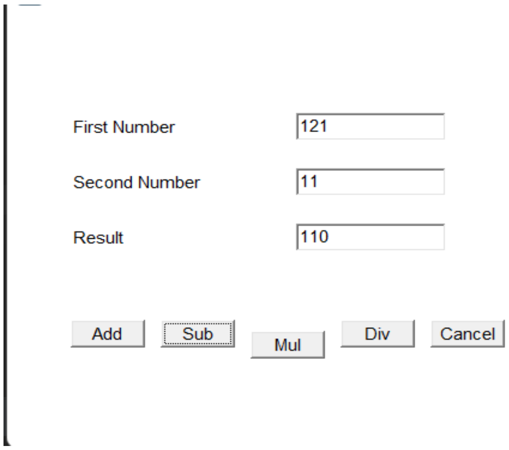


First Number 121

Second Number 11

Result 132

Substraction

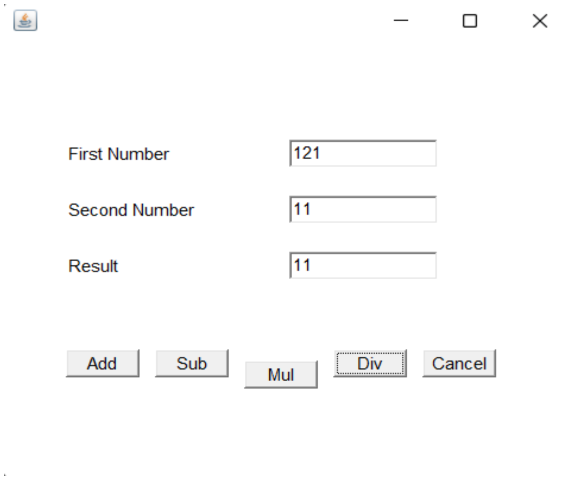


First Number 121

Second Number 11

Result 110

Division

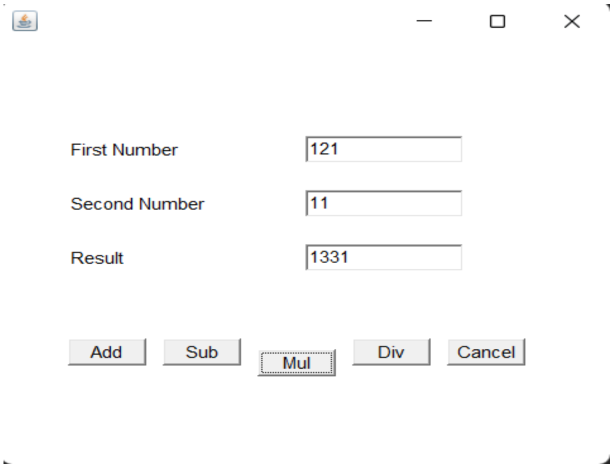


First Number 121

Second Number 11

Result 11

Multiplication



First Number 121

Second Number 11

Result 1331

CO6

EXPERIMENT 31

Aim

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

Program

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

public class EmployeeFileIO {
    public static void main(String[] args) {
        try {
            Scanner obj = new Scanner(System.in);
            FileOutputStream fout = new FileOutputStream("employee2.txt");
            int empno;
            String empname;
            int salary;
            System.out.println("Enter the number of employees:");
            int limit = obj.nextInt();
            obj.nextLine(); // Consume the newline left by nextInt()

            for (int j = 0; j < limit; j++) {
                try {
                    System.out.println("Enter the employee number:");
                    empno = obj.nextInt();
                    obj.nextLine(); // Consume the newline left by nextInt()
                    String a = String.valueOf(empno);
                    byte a1[] = a.getBytes();
                    fout.write(a1);
                    fout.flush();
                    fout.write("\r\n".getBytes());

                    System.out.println("Enter the employee name:");
                    empname = obj.nextLine();
                    byte b[] = empname.getBytes();
                    fout.write(b);
                    fout.flush();
                    fout.write("\r\n".getBytes());

                    System.out.println("Enter the employee salary:");
                    salary = obj.nextInt();
                    obj.nextLine(); // Consume the newline left by nextInt()
                    String c = String.valueOf(salary);
                    byte c1[] = c.getBytes();
                    fout.write(c1);
                    fout.flush();
                    fout.write("\r\n".getBytes());
                } catch (InputMismatchException e) {
                    System.out.println("Invalid input. Please enter a valid integer.");
                    obj.nextLine(); // Consume the invalid input
                    j--; // Decrement the loop counter to retry entering the employee's data
                }
            }
            fout.close();
        } catch (Exception e) {
            System.out.println(e);
        }

        try {
            FileInputStream fin = new FileInputStream("employee2.txt");
```

```
            System.out.println("Contents of the file:");
            int data;
            while ((data = fin.read()) != -1) {
                System.out.print((char) data);
            }
            fin.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Output

```
javac EmployeeFileIO.java
java EmployeeFileIO

Enter the number of employees:
2
Enter the employee number:
11
Enter the employee name:
Ajay
Enter the employee salary:
34300
Enter the employee number:
12
Enter the employee name:
Surya
Enter the employee salary:
43000
Contents of the file:
```

```
11
Ajay
34300
12
Surya
43000
```

EXPERIMENT 32

Aim

Write a program to copy one file to another.

Program

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

class CopyFile {
    public static void main(String arg[]) throws Exception {
        Scanner sc = new Scanner(System.in);
        System.out.print("Provide source file name: ");
        String sfile = sc.next();
        System.out.print("Provide destination file name: ");
        String dfile = sc.next();
        FileReader fin = new FileReader(sfile);
        FileWriter fout = new FileWriter(dfile, true); // "true" to append content to the destination file
        int c;
        while ((c = fin.read()) != -1) {
            fout.write(c);
        }
        System.out.println("Copy finish...");
        fin.close();
        fout.close();
    }
}
```

Output

source.txt

```
Java is a popular programming language, created in 1995.
It is owned by Oracle, and more than 3 billion devices run Java.
It is used for: Mobile applications (specially Android apps) Desktop applications
.
 javac CopyFile.java
 java CopyFile

Provide source file name: source.txt
Provide destination file name: destination.txt
Copy finish…
```

destination.txt

```
Java is a popular programming language, created in 1995.
It is owned by Oracle, and more than 3 billion devices run Java.
It is used for: Mobile applications (specially Android apps) Desktop applications
```

EXPERIMENT 33

Aim

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

Program

```
import java.io.*;
import java.util.Scanner;

public class EvenOddNumbers {
    public static void main(String[] args) {
        try {
            Scanner inputScanner = new Scanner(System.in);

            System.out.print("Enter the name of the input file: ");
            String inputFileName = inputScanner.nextLine();

            FileInputStream inputFile = new FileInputStream(inputFileName);
            Scanner fileScanner = new Scanner(inputFile);

            FileOutputStream evenFile = new FileOutputStream("even.txt");
            FileOutputStream oddFile = new FileOutputStream("odd.txt");

            PrintWriter evenWriter = new PrintWriter(evenFile);
            PrintWriter oddWriter = new PrintWriter(oddFile);
            while (fileScanner.hasNextInt()) {
                int number = fileScanner.nextInt();
                if (number % 2 == 0) {
                    evenWriter.println(number); // Write even numbers to the even file
                } else {
                    oddWriter.println(number);  // Write odd numbers to the odd file
                }
            }
            evenWriter.close();
            oddWriter.close();
            inputFile.close();
            System.out.println("Even and odd numbers have been copied to separate files.");
        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }
}
```

Output

```
number.txt
13,12,16,18,22,40,15,23
Javac EvenOdd.java
Java EvenOdd
Enter the name of the input file: numbers.txt
Even and odd numbers have been copied to separate files.
Odd.txt    even.txt
 13           10
 15           12
  23          14
              16
```