

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

LAB MANUAL

Academic Year : 2019 - 2020
Course Code : AITB06
Regulations : IARE - R18
Semester : IV
Branch : CSE

Prepared by

Mr. N V Krishna Rao

Assistant Professor

Mr. T Vijay Kumar

Assistant Professor

Ms. K Radhika

Assistant Professor



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

PROGRAM OUTCOMES (PO's)

PO-1	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems (Engineering Knowledge).
PO-2	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences (Problem Analysis).
PO-3	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations (Design/Development of Solutions).
PO-4	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions (Conduct Investigations of Complex Problems).
PO-5	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations (Modern Tool Usage).
PO-6	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice (The Engineer and Society).
PO-7	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development (Environment and Sustainability).
PO-8	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice (Ethics).
PO-9	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings (Individual and Team Work).
PO-10	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions (Communication).
PO-11	Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. (Project Management and Finance)
PO-12	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change (Life-long learning).

PROGRAM SPECIFIC OUTCOMES (PSO's)

PSO – I	Professional Skills: The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexity.
PSO – II	Problem-Solving Skills: The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.
PSO – III	Successful Career and Entrepreneurship: The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

ATTAINMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES

S. No	Experiment	Program Outcomes Attained	Program Specific Outcomes Attained
1.	BASIC PROGRAMS	PO1, PO2, PO5	PSO2
2.	MATRICES, OVERLOADING, OVERRIDING	PO1, PO2, PO5	PSO2
3.	PALINDROME, ABSTRACT CLASS	PO1, PO2, PO5	PSO2
4.	INTERFACE	PO1, PO2, PO5	PSO2
5.	MULTITHREADING	PO1, PO2, PO5	PSO2
6.	FILES	PO1, PO2, PO3, PO4, PO12	PSO1, PSO2
7	FILES	PO1, PO2, PO3, PO4, PO12	PSO1, PSO2
8.	JAVA PROGRAM WITH JDBC	PO1, PO2, PO3, PO4, PO12	PSO2
9.	FILES	PO1, PO2, PO3, PO4, PO12	PSO1, PSO2
10.	TRAFFIC LIGHT	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
11	MOUSE EVENTS	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
12	CALCULATOR	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
13	APPLETS	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
14	ADDITIONAL PROGRAMS	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

MAPPING COURSE OBJECTIVES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Objectives	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
I	√	√	√										√		
II	√	√		√	√								√		
III			√		√							√	√	√	
IV	√	√	√									√		√	

SYLLABUS:

OBJECT ORIENTED PROGRAMING THROUGH JAVA LABORATORY

IV Semester: CSE / IT								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
AITB06	Core	L	T	P	C	CIA	SEE	Total
		1	-	2	2	30	70	100
Contact Classes: 13		Tutorial Classes: Nil		Practical Classes: 26		Total Classes: 39		
OBJECTIVES:								
The course will enable the students to:								
I. Practice object-oriented programs and build java applications.								
II. Implement java programs for establishing interfaces.								
III. Implement sample programs for developing reusable software components.								
IV. Create database connectivity in java and implement GUI applications.								
LIST OF EXPERIMENTS								
Week-1	BASIC PROGRAMS							
a. Try debug step by step with small program of about 10 to 15 lines which contains at least one if else condition and a for loop.								
b. Write a java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula.								
c. The Fibonacci sequence is defined by the following rule. The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a java program that uses both recursive and non recursive functions.								
Week-2	MATRICES, OVERLOADING, OVERRIDING							
a. Write a java program to multiply two given matrices.								
b. Write a java program to implement method overloading and constructors overloading.								
c. Write a java program to implement method overriding.								
Week-3	PALINDROME, ABSTRACT CLASS							
a. Write a java program to check whether a given string is palindrome.								
b. Write a java program for sorting a given list of names in ascending order.								
c. Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.								
Week-4	INTERFACE							
Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.								

Week-5	MULTITHREADING
<p>a. Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.</p> <p>b. Write a java program that correct implements of producer consumer program.</p>	
Week-6	FILES
<p>a. Write a java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.</p> <p>b. Write a java program that displays the number of characters, lines and words in a text file.</p> <p>c. Write a java program that reads a file and displays the file on the screen with line number before each line.</p>	
Week-7	FILES
<p>a. Suppose that table named table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using labels in grid layout.</p> <p>b. Write a java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.</p>	
Week-8	JAVA PROGRAM WITH DATABASE
<p>a. Write a java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (/t). It takes a name or phone number as input and prints the corresponding other value from the hash table. Hint: Use hash tables.</p> <p>b. Implement the above program with database instead of a text file.</p>	
Week-9	FILES
<p>a. Write a java program that takes tab separated data (one record per line) from a text file and insert them into a database.</p> <p>b. Write a java program that prints the metadata of a given table.</p>	
Week-10	TRAFFIC LIGHT
<p>Write a java program that simulates a traffic light. The program lets the user select one of three lights: Red, Yellow or Green with radio buttons. On selecting a button an appropriate message with “STOP” or “READY” or ”GO” should appear above the buttons in selected color. Initially, there is no message shown.</p>	
Week-11	MOUSE EVENTS
<p>a. Write a java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. Use adapter classes.</p> <p>b. Write a java program to demonstrate the key event handlers.</p>	
Week-12	CALCULATOR
<p>Write a java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result. Handle any possible exception like divided by zero.</p>	

Week-13**APPLET**

- a. Develop an applet that displays a simple message.
- b. Develop an applet that receives an integer in one text field and computes its factorial value and returns it in another text field, when the button named “compute” is clicked.

ADDITIONAL PROGRAMS

1. Design a class to represent a Student details include the Student ID, Name of the Student, Branch, year, location and college. Assign initial values using constructor. Calculate average of marks of 6 subjects and calculate attendance percentage.
2. Develop with suitable hierarchy, classes for Point, Shape, Rectangle, Square, Circle, Ellipse, Triangle, Polygon, etc. Design a simple test application to demonstrate dynamic polymorphism
3. Write a program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.
4. Write an Applet that computes the payment of a loan on the amount of the loan, the interest rate and the number of months. It takes one parameter from the browser. Monthly; if true, the interest rate is per month, otherwise the interest rate is annual.
5. Write a java programs to find factorial of a number. user is allowed to enter a number into the text field whose factorial is to be determined. On pressing the button the value of the text field is firstly converted into integer and then processed to find its factorial. The result will get display in another text field.(Hint: use swings).

Reference Books:

1. P. J. Deitel, H. M. Deitel, “Java for Programmers”, Pearson Education, PHI, 4th Edition, 2007.
2. P. Radha Krishna, “Object Oriented Programming through Java”, Universities Press, 2nd Edition, 2007
3. Bruce Eckel, “Thinking in Java”, Pearson Education, 4th Edition, 2006.
4. Sachin Malhotra, Saurabh Chaudhary, “Programming in Java”, Oxford University Press, 5th Edition, 2010.

Web References:

1. http://vlab.co.in/ba_labs_all.php?id=2
2. <http://www.javatpoint.com/java-programs>
3. <http://introc.cs.princeton.edu/java/10elements/>

INDEX

S.NO	LIST OF EXPERIMENTS	PAGE NO
BASIC PROGRAMS		
WEEK-1	1.	Try debug step by step with small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
	2.	Write a java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula.
	3.	The Fibonacci sequence is defined by the following rule. The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a java program that uses both recursive and non recursive functions.
MATRICES, OVERLOADING, OVERRIDING		
WEEK-2	1.	a. Write a java program to multiply two given matrices.
	2.	b. Write a java program to implement method overloading and constructors overloading.
	3.	c. Write a java program to implement method overriding.
PALINDROME, ABSTRACT CLASS		
WEEK-3	1.	Write a java program whether a given String is palindrome or not
	2.	Write a java program for sorting a given list of names in ascending order.
	3.	Write a java program to create an abstract class named Shape that contains two integers and an empty method named printArea (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea () that prints the area of the given shape.
INTERFACE		
WEEK-4	1	Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

	MULTITHREADING	
WEEK-5	1	Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
	2	Write a java program that implements producer consumer program.
	FILES	
WEEK-6	1	Write a java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
	2	Write a java program that displays the number of characters, lines and words in a text file.
	3	Write a java program that reads a file and displays the file on the screen with line number before each line.
	FILES AND JAVA PROGRAM WITH JDBC	
WEEK-7 and WEEK-8	1	Suppose that table named table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using labels in grid layout.
	2	Write a java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.
	1	Write a java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (/t). It takes a name or phone number as input and prints the corresponding other value from the hash table. Hint: Use hash tables.
	2	Implement the above program with database instead of a text file.
WEEK-9	1	Write a java program that takes tab separated data (one record per line) from a text file and insert them into a database.
	2	Write a java program that prints the metadata of a given table.
	FILES (EVENT HANDLING, AWT)	
WEEK-10	1	Write a java program that simulates a traffic light. The program lets the user select one of three lights: Red, Yellow or Green with radio buttons. On selecting a button an appropriate message with "STOP" or "READY" or "GO" should appear above the buttons in selected color. Initially, there is no message shown.

WEEK-11	1. Write a java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. Use adapter classes.	
	2. Write a java program to demonstrate the key event handlers.	
WEEK-12	CALCULATOR (EVENT HANDLING AND LAYOUTS)	
	1. Write a java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result. Handle any possible exception like divided by zero.	
WEEK-13	APPLETS	
	1. Develop an applet that displays a simple message.	
	2. Develop an applet that receives an integer in one text field and computes its factorial value and returns it in another text field, when the button named "compute" is clicked.(Hint: Applet).	
	ADDITIONAL PROGRAMS	
	1. Design a class to represent a Student details include the Student ID, Name of the Student, Branch, year, location and college. Assign initial values using Constructor. Calculate average of marks of 6 subjects and calculate attendance percentage.	
	2. Develop with suitable hierarchy, classes for Point, Shape, Rectangle, Square, Circle, Ellipse, Triangle, Polygon, etc. Design a simple test application to demonstrate dynamic polymorphism.	
	3. Write an Applet that computes the payment of a loan on the amount of the loan, the interest rate and the number of months. It takes one parameter from the browser. Monthly; if true, the interest rate is per month, otherwise the interest rate is annual.	
	4. Write a program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.	
	5. Write a java programs to find factorial of a number. user is allowed to enter a number into the text field whose factorial is to be determined. On pressing the button the value of the text field is firstly converted into integer and then processed to find its factorial. The result will get displayed in another text field.(Hint: use swings).	

WEEK-1: BASIC PROGRAMS

EXPERIMENT-1.1

OBJECTIVE:

To debug step by step with small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

RESOURCES:

Net Beans 8.0.2, JDK 7.0 is required, 1GB RAM, Hard Disk 80 GB.

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name „Testp“.
3. Read n values from keyboard using a Scanner class.
4. Find the (n % i == 0) using dynamic initialization
5. Check the (n % i == 0) value using if condition.

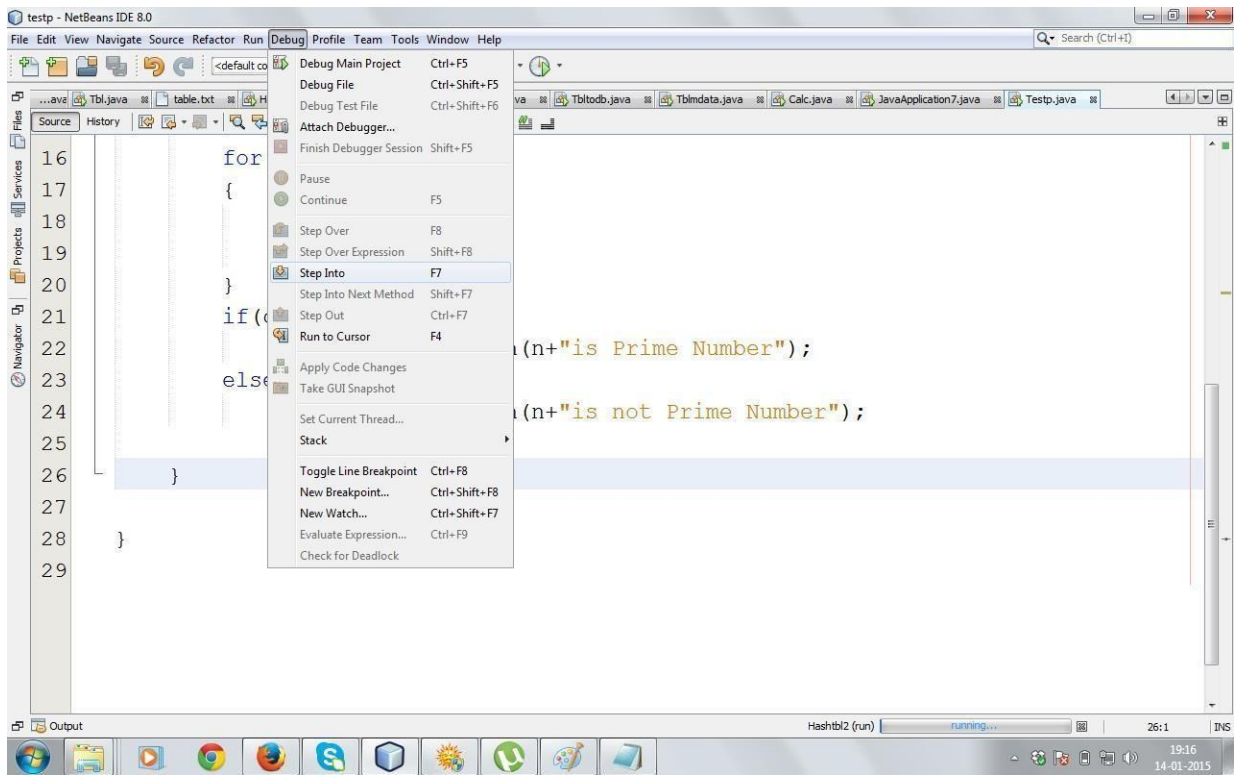
SOURCE CODE:

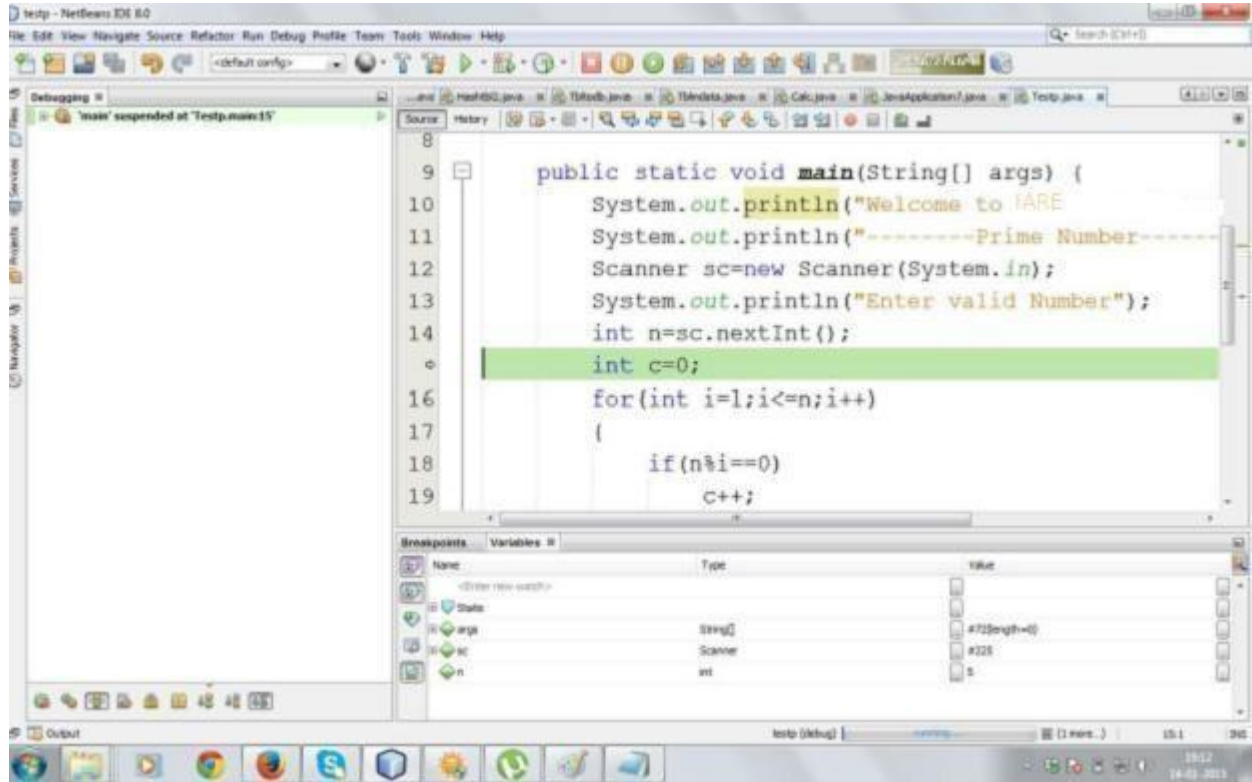
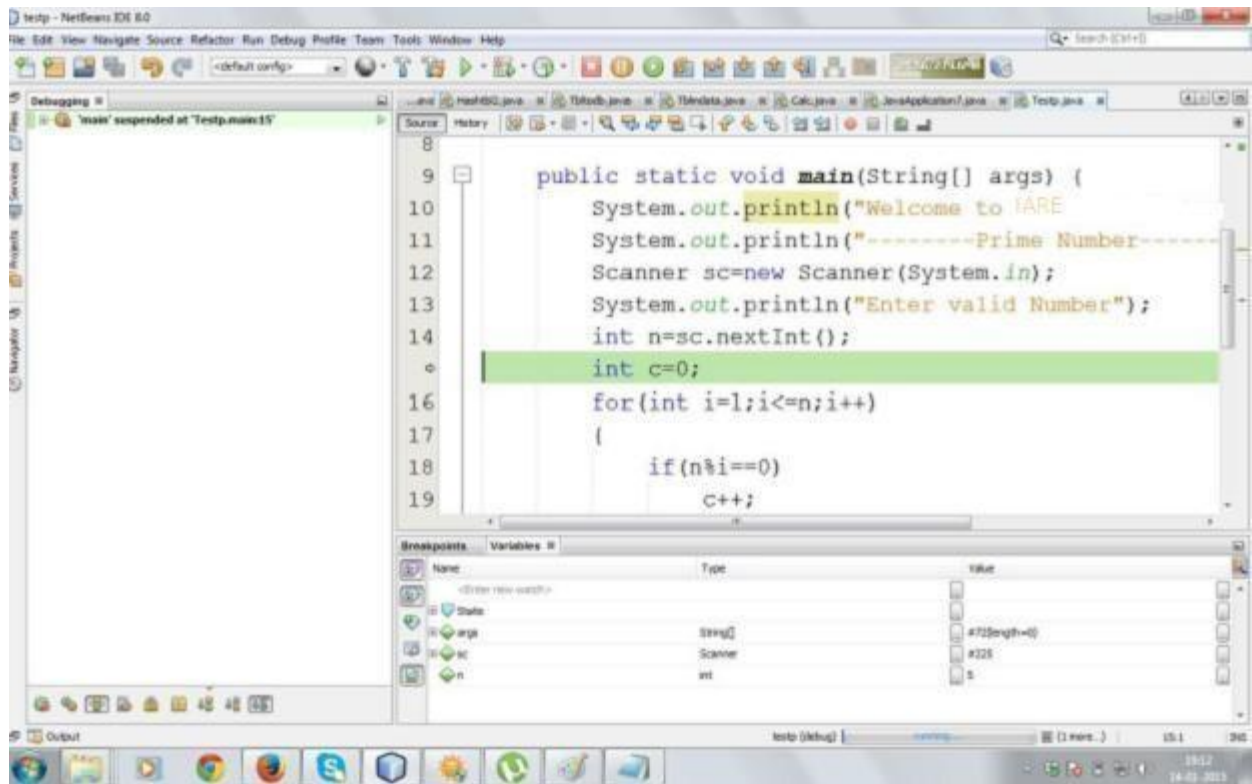
```
import java.util.*;

public class Testp {
    public static void main(String[] args) {
        System.out.println("Welcome to Institute of Engineering College");
        System.out.println("-----Prime Number -----");
        Scanner sc = new scanner(System.in);
        System.out.println("Enter valid Number");
        int n = sc.nextInt();
        int c = 0;
        for (int i = 1; i <= n; i++)
        {
            if (n % i == 0)
            {
                c++;
            }
        }
        if (c == 2)
        {
            System.out.println(n + "is Prime Number");
        }
        else
            System.out.println(n + "is not Prime Number");

    }
}
```

OUTPUT:





PRE LAB VIVA QUESTIONS:

1. Difference between header files and import packages?
2. What is the difference between procedural and object-oriented programs?
3. What are Encapsulation, Inheritance and Polymorphism?
4. What is OOPs?

POST LAB VIVA QUESTIONS:

5. What is an object and how do you allocate memory to it?
6. What is the difference between an argument and a parameter?
7. What are different types of access modifiers?
8. What is unicode?

EXPERIMENT-1.2

OBJECTIVE:

Write a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name „roots“.
3. Declare the double variables r1, r2.
4. Read a, b, c values from keyboard using a Scanner class.
5. Find the discriminant value using dynamic initialization
`int d=b*b-4*a*c;`
6. Check the discriminant value using if condition.

COMPIATION:

To execute a java program we require setting a class path:

```
C:\set path= C:\Program Files\Java\jdk1.6.0\bin,;
```

```
C:\javac roots.java
```

EXCEUTION:

```
C:\java roots
```

SOURCE CODE:

```
import java.util.*;
class roots
{
    public static void main(String args[])
    {
        double r1,r2;
        System.out.println("Enter a,b,c of the equation ax2+bx+c=0");
        Scanner as=new Scanner (System.in);
        int a=as.nextInt();
        int b=as.nextInt();
        int c=as.nextInt();
        int d=b*b-4*a*c;
        if(d<0)
        {
            System.out.println("There are no solutions");
        }
    }
}
```



```

    }
    if(d==0)
    {
        System.out.println("The roots are real and equal");
        r1=r2=-b/(2.0*a);
        System.out.println("The roots are "+r1+" and "+r2);
    }
    if(d>0)
    {
        System.out.println("The roots are real");
    }
    if(d<0)
    {
        System.out.println("The roots are real");
        r1=(-b+Math.sqrt(d))/(2.0*a);
        r2=(-b-Math.sqrt(d))/(2.0*a);
        System.out.println("The roots are "+r1+" and "+r2);
    }
}
}

```

OUTPUT:

```

C:\WINDOWS\system32\cmd.exe
I:\java\week 1>javac Quad.java
I:\java\week 1>java Quad 1 4 4
-2.0 -2.0
I:\java\week 1>_

```

PRE LAB VIVA QUESTIONS:

1. What is an array?
2. Define class with example?
3. Define object?
4. Define oop?

POST LAB VIVA QUESTIONS:

1. Define variable?
2. What is instance variable?
3. Define class variable?
4. What is an expression?

EXPERIMENT-1.3

OBJECTIVE:

The Fibonacci sequence is defined by the following rule: The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a Java program that uses both recursive and non recursive functions to print the nth value in the Fibonacci sequence.

RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name „Fibnon“.
3. Declare the int variables n, n1.
4. Create an object for predefined class Random and generate a random using nextInt()
5. In main() create an object for fib1 class and call the method fibc() using the object.

COMPLIATION:

To execute a java program we require setting a class path:

```
C:\set path= C:\Program Files\Java\jdk1.6.0\bin; ;
```

```
C:\javac Fibnon.java
```

EXCEUTION:

```
C:\java Fibnon
```

SOURCE CODE:

```
import java.util.*;
class Fibnon
{
    public static void main(String args[])
    {
        int n,n1;
        Random as=new Random();
        n=as.nextInt(10);
        fib1 as1=new fib1 ();
        n1=as1.fibc(n);
        System.out.println("The "+n+"th fibonacci number is "+n1);
    }
}
```

```

class fib1
{
    int a=1,b=1;
    int fib;
    int fibc(int n)
    {
        System.out.print(a+" "+b+" ");
        for(int i=1;i<=n-2;i++)
        {
            fib=a+b;
            System.out.print(fib+" ");
            a=b;
            b=fib;
        }
        return fib;
    }
}

```

OUTPUT:

```

C:\WINDOWS\system32\cmd.exe
I:\java\week 1>javac Fibo.java
I:\java\week 1>java Fibo 9
0
1
1
2
3
5
8
13
21
34
Fibonacci series using recursion
1
1
2
3
5
8
13
21
34

```

Or

```

import java.util.Scanner;
import java.io.*;
class Fibonacci
{
    int a1,a2,a3,num,i;

```

```

int fibRec(int x)
{
    if(x==0)
        return 0;
    else if(x==1)
        return 1;
    else
        return (fibRec(x-1)+fibRec(x-2));
}
int fibNonRec(int y)
{
    a1=1;
    a2=1;
    num=y;
    for(i=3;i<=num;i++)
    {
        a3=a1+a2;
        System.out.println(a3);
        a1=a2;
        a2=a3;
    }
    return a3;
}
}
class Fib
{
    public static void main(String args[]) throws IOException
    {
        System.out.println("enter n value");
        Scanner s = new Scanner(System.in);
        int n = s.nextInt();

        Fibonacci f=new Fibonacci();
        for (int i = 0; i <= n; i++) {
            System.out.print(f.fibRec(i) + " ");
        }
        System.out.println("The nth value with recursive:"+f.fibRec(n));
        System.out.println("The nth value with non-recursive:"+f.fibNonRec(n));
    }
}

```

Compile:javac Fib.java
Run:java Fib

OUTPUT:
enter n value
5
0 1 1 2 3 5
The nth value with recursive:5
2

3
5

The nth value with non-recursive: 5

PRE LAB VIVA QUESTIONS:

1. Define recursion?
2. How call by reference is done?
3. Describe object creation?
4. Describe class and its objects?

POST LAB VIVA QUESTIONS:

1. Define method?
2. How method is called?
3. How static methods are called?
4. Define constructor?

EXPERIMENT-1.4

OBJECTIVE:

Read array of numbers through command line and sort in ascending order.

RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Create a class and pass array numbers through command line arguments.
2. And give these values to array.
3. And sort the array using any sorting methods and print the array.

COMPLIATION:

To execute a java program we require setting a class path:

```
C:\set path= C:\Program Files\Java\jdk1.6.0\bin.;
```

```
C:\javac JavaApplication2.java
```

EXCEUTION:

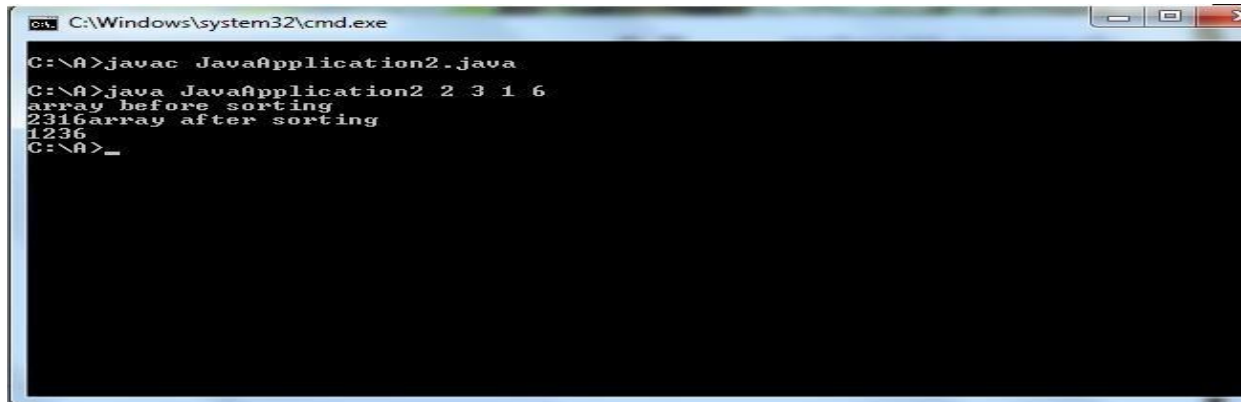
```
C:\java JavaApplication2
```

SOURCE CODE:

```
public class JavaApplication2
{
    // @param args the command line arguments
    public static void main(String[] args)
    {
        // TODO code application logic here
        // dataType[] arrayRefVar = new dataType[arraySize]
        int[] a=new int[10];
        for(int i = 0; i < args.length; i++)
        {
```

```
        a[i]=Integer.parseInt(args[i]);
    }
    System.out.println("array before sorting");
    for(int i = 0; i < args.length; i++)
    {
        System.out.print(a[i]);
    }
    for(int i = 0; i < args.length; i++)
    {
        for ( int k = i+1;k <args.length;k++)
        {
            if(a[i]>a[k])
            {
                int temp;
                temp=a[i];
                a[i]=a[k];
                a[k]=a[i];
            }
        }
    }
    System.out.println("array after sorting");
    for(int i=0;i<args.length;i++)
    System.out.print(a[i]);
}
}
```


OUTPUT:



```
C:\Windows\system32\cmd.exe
C:\A>javac JavaApplication2.java
C:\A>java JavaApplication2 2 3 1 6
array before sorting
2316array after sorting
1236
C:\A>_
```

PRE LAB VIVA QUESTIONS:

1. Difference between header files and import packages?
2. What is the difference between procedural and object-oriented programs?
3. What are Encapsulation, Inheritance and Polymorphism?
4. What is OOPs?

POST LAB VIVA QUESTIONS:

1. What is an object and how do you allocate memory to it?
2. What is the difference between an argument and a parameter?
3. What are different types of access modifiers?
4. What is unicode representation?

WEEK-2: POLYMORPHISM, INHERITANCE

EXPERIMENT-2.1

1. Write a java program for matrix multiplication.

```
1. // This is sample program for matrix multiplication
2. // The complexity of the algorithm is  $O(n^3)$ 
3. package com.sanfounry.numerical;
4.
5. import java.util.Scanner;
6.
7. public class MatixMultiplication
8. {
9.     public static void main(String args[])
10.    {
11.        int n;
12.        Scanner input = new Scanner(System.in);
13.        System.out.println("Enter the base of squared matrices");
14.        n = input.nextInt();
15.        int[][] a = new int[n][n];
16.        int[][] b = new int[n][n];
17.        int[][] c = new int[n][n];
18.        System.out.println("Enter the elements of 1st martix row wise \n");
19.        for (int i = 0; i < n; i++)
20.        {
21.            for (int j = 0; j < n; j++)
22.            {
23.                a[i][j] = input.nextInt();
24.            }
25.        }
26.        System.out.println("Enter the elements of 2nd martix row wise \n");
27.        for (int i = 0; i < n; i++)
28.        {
29.            for (int j = 0; j < n; j++)
30.            {
31.                b[i][j] = input.nextInt();
32.            }
33.        }
34.        System.out.println("Multiplying the matrices...");
35.        for (int i = 0; i < n; i++)
36.        {
37.            for (int j = 0; j < n; j++)
38.            {
39.                for (int k = 0; k < n; k++)
40.                {
41.                    c[i][j] = c[i][j] + a[i][k] * b[k][j];
42.                }
43.            }
44.        }
45.        System.out.println("The product is:");
46.        for (int i = 0; i < n; i++)
47.        {
```

```
48.     for (int j = 0; j < n; j++)
49.     {
50.         System.out.print(c[i][j] + " ");
51.     }
52.     System.out.println();
53. }
54. input.close();
55. }
56. }
```

Output:

```
$ javac MatixMultiplication.java
$ java MatixMultiplication
```

Enter the base of squared matrices:
3

Enter the elements of 1st martix row wise:
1 2 3
4 5 6
7 8 9

Enter the elements of 2nd martix row wise:
2 3 4
5 6 7
8 9 1

Multiplying the matrices...
The product is:
36 42 21
81 96 57
126 150 93

EXPERIMENT-2.2

2. Design a class to represent a bank account. Which include contains account number, name of the depositor, type of the account, balance amount in the account. Use constructors to assign initial values, to Deposit an amount, to Withdraw amount after checking balance, to display name and balance. (Hint: constructor overloading).

RESOURCES:

Net Beans 8.0.2, JDK 7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Create a class for bank account.
2. Declare account number, name, type of account, balance as class members.
3. Define constructor to assign initial values.
4. Define methods to deposit an amount, to withdraw an amount.
5. While with drawing an amount check the balance.

SOURCE CODE:

```
import java.util.Scanner;

class BankAccount
{
    int acno;
    String name;
    String type;
    int bal;

    BankAccount(int a,String b,String c,int d)
    {
        acno=a;
        name=b;
        type=c;
```

```
        bal=d;
    }

void Deposit(int a)
{
    System.out.println("Balance before deposit is"+bal);
    bal=bal+a;
    System.out.println("Balance after deposit is"+bal);
}

void Withdraw(int a)
{
    System.out.println("Balance before withdrawal is"+bal);
    bal=bal-a;
    if(bal<0)
    {
        System.out.println("cannot withdraw");
        bal=bal+a;
    }
    else
        System.out.println("Balance after withdrawal is"+bal);
}

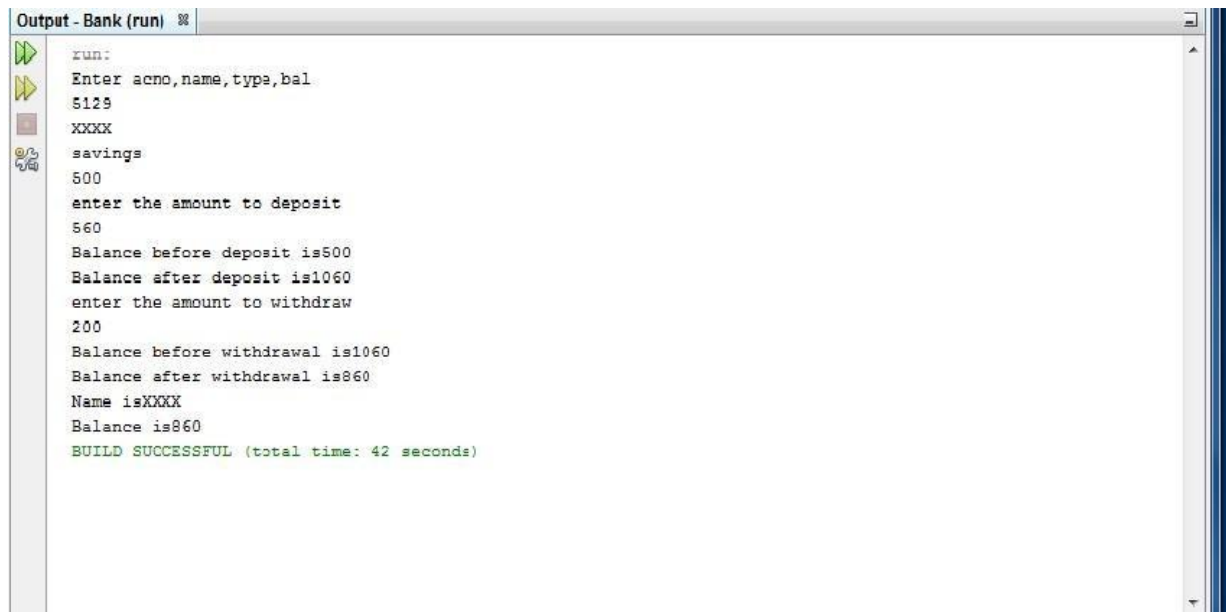
void Display()
{
    System.out.println("Name is"+name);
    System.out.println("Balance is"+bal);
}
}
```

```

class Bank1
{
    public static void main(String args[])
    {
        BankAccount ba=new BankAccount();
        Scanner s=new Scanner(System.in);
        System.out.println("Enter acno,name,type,bal");
        ba.BankAccount(s.nextInt(),s.next(),s.next(),s.nextInt());
        System.out.println("enter the amount to deposit");
        ba.Deposit(s.nextInt());
        System.out.println("enter the amount to withdraw");
        ba.Withdraw(s.nextInt());
        ba.Display();
    }
}

```

OUTPUT:



```

Output - Bank (run)
run:
Enter acno,name,type,bal
5129
XXXX
savings
500
enter the amount to deposit
560
Balance before deposit is500
Balance after deposit is1060
enter the amount to withdraw
200
Balance before withdrawal is1060
Balance after withdrawal is860
Name isXXXX
Balance is860
BUILD SUCCESSFUL (total time: 42 seconds)

```

PRE LAB VIVA QUESTIONS:

1. Describe constructors with parameter and without parameter?
2. Define scanner class?
3. What is the method to read string?
4. What is the method to int, char, float?

Post LAB VIVA QUESTIONS:

1. Describe string methods?
2. What is string buffer class?
3. String class is defined in which package?
4. List String Tokenizer methods?

EXPERIMENT-2.2

OBJECTIVE:

Write a java program for method overloading.

RESOURCES:

Net Beans 8.0.2, JDK 7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Create a class OverloadDemo.
2. Create three methods with the same name but different in their parameter list.
3. Create main class.
4. Create object of OverloadDemo and call the methods.

SOURCE CODE

```
1. class OverloadDemo
2. {
3.     void area(float x)
4.     {
5.         System.out.println("the area of the square is "+Math.pow(x, 2)+" sq units");
6.     }
7.     void area(float x, float y)
8.     {
9.         System.out.println("the area of the rectangle is "+x*y+" sq units");
10.    }
11.    void area(double x)
12.    {
13.        double z = 3.14 * x * x;
14.        System.out.println("the area of the circle is "+z+" sq units");
15.    }
16. }
17. class Overload
18. {
19.     public static void main(String args[])
20.     {
21.         OverloadDemo ob = new OverloadDemo();
22.         ob.area(5);
23.         ob.area(11,12);
24.         ob.area(2.5);
25.     }
26. }
```

Output:

```
$ javac OverloadDemo.java
$ java OverloadDemo
```

```
the area of the square is 25.0 sq units
the area of the rectangle is 132.0 sq units
the area of the circle is 19.625 sq units
```


PRE LAB VIVA QUESTIONS

- 1 What is method overloading?
- 2 Can same methods present in different classes?
- 3 How to call methods?
- 4 Describe final class?

POST LAB VIVA QUESTIONS

- 1 What is garbage collector?
- 2 Define final variable?
- 3 Define finalize method?
- 4 Define inheritance?

EXPERIMENT-2.3

OBJECTIVE:

Write a java program to implement method overriding.

RESOURCES:

Net Beans 8.0.2, JDK 7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Create a class MODemo.
2. Create a subclass B extends MODemo..
3. Subclass has same method has super class.
4. Create object to call methods.

SOURCE CODE:

```
// Method overriding.
class MODemo
{
    int i, j;
    MODemo (int a, int b)
    {
        i = a;
        j = b;
    }
    // display i and j
    void show()
    {
        System.out.println("i and j: " + i + " " + j);
    }
}
class B extends MODemo
{
    int k;
    B(int a, int b, int c)
    {
        super(a,b);
        k = c;
    }
    // display k – this overrides show() in A
    void show()
    {
        System.out.println("k: " + k);
    }
}
```

```
class Override
{
    public static void main(String args[])
    {
        B subOb = new B(1, 2, 3);
        subOb.show(); // this calls show() in B
    }
}
```

OUTPUT:

i and j:1 2

k: 3

PRE LAB VIVA QUESTIONS:

1. What is method overriding?
2. What is subclass?
3. What is superclass?
4. What is method overloading?

Post LAB VIVA QUESTIONS:

5. Describe how subclass methods override super class?
6. When we use method overriding?
7. Which object is used to call super class members?
8. Difference between method overloading and method overriding?

WEEK-3: PALINDROME AND ABSTRACT CLASS

OBJECTIVE:

Experiment 3.1

Write a java program to check whether a given string is palindrome.

```
import java.util.Scanner;

class ChkPalindrome
{
    public static void main(String args[])
    {
        String str, rev = "";
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a string:");
        str = sc.nextLine();

        int length = str.length();

        for ( int i = length - 1; i >= 0; i-- )
            rev = rev + str.charAt(i);

        if (str.equals(rev))
            System.out.println(str+" is a palindrome");
        else
            System.out.println(str+" is not a palindrome");

    }
}
```

Output:

```
MADAM
Is a palindrome
```

OBJECTIVE:

Experiment 3.2

Write a java program for sorting a given list of names in ascending order.

RESOURCES:

Net Beans 8.0.2, JDK 7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name Sorting.
3. Create a method called sortStrings().
4. Create the objects of the Sorting class.
5. Call the sortStrings() method.

```
import java.util.*;
class Sorting
{
    void sortStrings()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the value of n: ");
        int n = s.nextInt();
        String[] str = new String[n];
        System.out.println("Enter strings: ");
        for(int i = 0; i < n; i++)
        {
            str[i] = new String(s.next());
        }
        for(int i = 0; i < n; i++)
        {
            for(int j = i+1; j < n; j++)
            {
                if(str[i].compareTo(str[j])>0)
                {
                    String temp = str[i];
                    str[i] = str[j];
                    str[j] = temp;
                }
            }
        }
    }
}
```

```
        System.out.println("Sorted list of strings is:");
        for(int i = 0; i < n ; i++)
        {
            System.out.println(str[i]);
        }
    }
}
public class Driver
{
    public static void main(String[] args)
    {
        Sorting obj = new Sorting();
        obj.sortStrings();
    }
}
```

Input and output for the above program are as follows:

Enter the value of n:

4

Enter strings:

suresh
mahesh
dinesh
ganesh

Sorted list of strings is:

dinesh
ganesh
mahesh
suresh

Experiment 3.3

To create an abstract class named shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle subclass that each one of the classes extends the Class Shape. Each one of the classes contains only the method printArea() that prints the area of Shape.

RESOURCES:

Net Beans 8.0.2, JDK 7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

6. Start the program
7. Create a class with the name Shape.
8. Create the Rectangle, Triangle, Circle extends Shape.
9. Create the objects to the individual classes.
10. Call the PrintArea() on individual object.

SOURCE CODE:

```
abstract class shape
{
    public int x, y;
    public abstract void printArea();
}
class Rectangle extends shape
{
    public void printArea()
    {
        System.out.println("Area of Rectangle is " + x * y);
    }
}
class Triangle extends shape
{
    public void printArea()
    {
        System.out.println("Area of Triangle is " + (x * y) / 2);
    }
}
class Circle extends shape
{
    public void printArea()
    {
        System.out.println("Area of Circle is " + (22 * x * x) / 7);
    }
}
```

```

public class Abstex
{
    public static void main(String[] args)
    {
        // TODO code application logic here

        Rectangle r = new Rectangle();
        r.x = 10;
        r.y = 20;
        r.printArea();
        System.out.println("----- ");

        Triangle t = new Triangle();
        t.x = 30;
        t.y = 35;
        t.printArea();
        System.out.println("----- ");

        Circle c = new Circle();
        c.x = 2;
        c.printArea();
        System.out.println("----- ");
    }
}

```

INPUT AND OUTPUT:

```

run:
Area of Rectangle is 200
-----
Area of Triangle is 525
-----
Area of Circle is 12
-----
BUILD SUCCESSFUL (total time: 3 seconds)
|

```

PRE LAB VIVA QUESTIONS:

1. What is an interface?
2. What is an abstract class?
3. Where abstract class is used?
4. Can we create an object of abstract class?

POST LAB VIVA QUESTIONS:

1. What is difference between abstract class and interface?
2. Can a abstract class be defined without any abstract methods?
3. Can an interface extend another interface?
4. Can an abstract class extend another interface?

EXPERIMENT 3.2

OBJECTIVE:

Write a java program to create an interface named Shape that contains empty method named Area () and perimeter(). Provide two classes named Triangle and Circle such that each one of the classes implement Shape. Each one of the classes contains only the method Area () and perimeter().that prints the area and perimeter() of the given shape.

RESOURCES:

Net Beans 8.0.2, JDK 7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name Shape.
3. Create the Triangle,Circle implements Shape.
4. Create the objects to the individual classes.
5. Call the Area() and Perimeter() on individual object.

SOURCE CODE:

```
package pkgif;
interface Shape
{
    double PI=3.14;
    void area();
    void perimeter();
}
interface Showable extends Shape
{
    void show();
}
abstract class Circle implements Shape
{
    double r;
    Circle(double r)
    {
        this.r=r;
    }
    public void area()
    {
        System.out.println("area of circle"+PI*r*r);
    }
}
```

```
class Circle1 extends Circle implements Shape, Showable
```

```
{  
    Circle1(double r)  
    {  
        super(r);  
    }  
    public void area()  
    {  
        System.out.println("area of circle"+PI*r*r);  
    }  
    public void perimeter()  
    {  
        System.out.println("perimeter of circle"+2*PI*r);  
    }  
    public void show()  
    {  
        System.out.println("*****This is circle*****");  
    }  
    void display()  
    {  
        System.out.println("PI value"+Shape.PI);  
    }  
}
```

```
class Triangle implements Showable
```

```
{  
double a,b,c;  
Triangle(double a,double b,double c)  
{  
    this.a=a;  
    this.b=b;  
    this.c=c;  
}  
  
public void area()  
{  
    System.out.println("area of Triangle"+a*b/2);  
}  
public void perimeter()  
{  
    System.out.println("perimeter of Triangle"+a*b*c);  
}  
public void show()  
{
```

```

        System.out.println("****This is triangle****");
    }
}

public class If {
    public static void main(String[] args) {
        System.out.println("****Implementing circle ****");
        Circle1 c=new Circle1(10.5);
        c.area();
        c.perimeter();
        c.show();
        c.display();

        System.out.println("****by reference variable****");
        Shape s;

        s=(Shape) c;//Explicit type casting is done because circle1 class is higher than shape interface

        s.area();

        s.perimeter();

        // s.display(); not possible to represent this statment because reference variables refers only abstract
        methods in the interface.

        System.out.println("****Implementing Triangle ****");

        Triangle t=new Triangle(10.5,2.2,3.3);

        t.area();

        t.perimeter();

        t.show();

    }
}

```

OUTPUT:

```

/* ****Implementing circle ****
area of circle346.185
perimeter of circle65.94
****This is circle****

```

PI value3.14

by reference variable

area of circle346.185

perimeter of circle65.94

****Implementing Triangle ****

area of Triangle11.55

perimeter of Triangle76.23

****This is triangle****

BUILD SUCCESSFUL (total time: 0 seconds) */

PRE LAB VIVA QUESTIONS:

1. What is an interface creation?
2. What is an abstract class vs. interface?
3. When abstract class is used?
4. Can we create an object of interface class?

POST LAB VIVA QUESTIONS:

5. What is difference between class and interface?
6. Can a interface class be defined without any abstract methods?
7. Can an interface implement another interface?
8. Can an abstract class extend another abstract class?

WEEK-4: INTERFACES

EXPERIMENT 4.1

OBJECTIVE:

You will be given two integers x and y as input, you have to compute x/y. Implement a class which raise an exception if x and y are not signed integers or if y is zero.

RESOURCES:

Net Beans 8.0.2, JDK 7.0 is required, 1GB RAM, Hard Disk 80 GB .

PROGRAM LOGIC:

1. Create user defined exception class named as NumberException, which extends Exception class.
2. In Number Exception class print the statement exception due to crossed 32 bit.
3. Create a class Throw Exception which contains a static method and throws the NumberException.
4. This method read two arguments, define a method such that it will check the range of numbers is within the 32 bit range or not. if not throw the exception.

SOURCE CODE:

```
package javaapplication1;
import java.util.Scanner;

class NumberException extends Exception
{
    NumberException()
    {
        System.out.println("exception due to crossed 32 bit");
    }
}
class ThrowException
{
    static void exception(int x1,int y1)throws NumberException
    {
        if(x1> 2147483647||y1>2147483647)
            throw new NumberException();
    }
}
class BitException
{
    public static void main(String ars[])
    {
        try
        {
```

```

int x,y;
float z;
Scanner s=new Scanner(System.in);
System.out.println("enter x and y values");
x=s.nextInt();
y=s.nextInt();
ThrowException.exception(x, y);

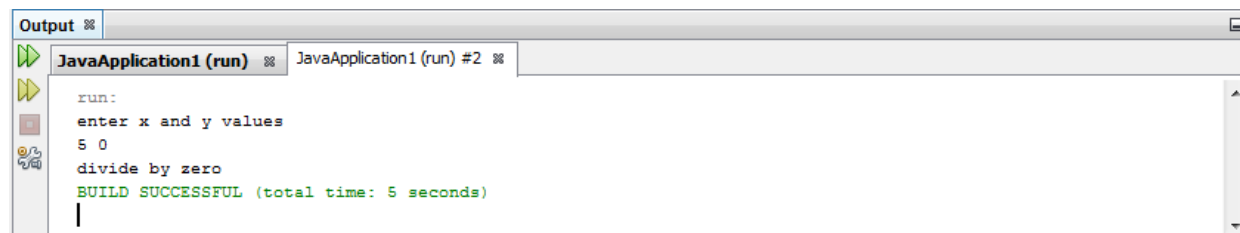
z=x/y;
System.out.println("result="+z);

}
catch(NumberException e)
{

}
catch(ArithmeticException e)
{
    System.out.println("divide by zero");
}
}
}

```

OUTPUT:



```

Output
JavaApplication1 (run) JavaApplication1 (run) #2
run:
enter x and y values
5 0
divide by zero
BUILD SUCCESSFUL (total time: 5 seconds)
|

```

PRE LAB VIVA QUESTIONS:

1. What is an Exception?
2. What is a Runtime Exception?
3. What is try block?
4. What is catch block?

POST LAB VIVA QUESTIONS:

5. What is difference between error and exception?
6. What is throwable class?
7. What is checked exception?
8. What is unchecked exception?

EXPERIMENT 4.2

OBJECTIVE:

Write a java program on user defined exceptions.

RESOURCES:

Net Beans 8.0.2, JDK 7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Create user defined exception class named as UserDefinedException.
2. In Account class, which is representing a bank account where you can deposit and withdraw money.
3. To withdraw money which exceeds your bank balance? You will not be allowed, and this is where user defined exception NotSufficientFundException is displayed.

SOURCE CODE:

/*Here we have an Account class, which is representing a bank account where you can deposit and withdraw money, but what will happen if you want to withdraw money which exceeds your bank balance? You will not be allowed, and this is where user defined exception comes into the picture. We have created a custom exception called NotSufficientFundException to handle this scenario. This will help you to show a more meaningful message to user and programmer. */

```
public class UserDefinedException {

    public static void main(String args[]) {
        Account acct = new Account();
        System.out.println("Current balance : " + acct.balance());
        System.out.println("Withdrawing $200");
        acct.withdraw(200);
        System.out.println("Current balance : " + acct.balance());
        acct.withdraw(1000);
    }

}

/**
 * Java class to represent a Bank account which holds your balance and provide wi
 */
public class Account {

    private int balance = 1000;
```

```

public int balance() {
    return balance;
}

public void withdraw(int amount) throws NotSufficientFundException {
    if (amount > balance) {
        throw new NotSufficientFundException(String.format("Current balance %d is less than requested
amount %d", balance, amount));
    }
    balance = balance - amount;
}

public void deposit(int amount) {
    if (amount <= 0) {
        throw new IllegalArgumentException(String.format("Invalid deposit amount %s", amount));
    }
}
}

public class NotSufficientFundException extends RuntimeException {

    private String message;

    public NotSufficientFundException(String message) {
        this.message = message;
    }

    public NotSufficientFundException(Throwable cause, String message) {
        super(cause);
        this.message = message;
    }

    public String getMessage() {
        return message;
    }
}

```

OUTPUT: Current balance : 1000
 Withdrawing \$200 Current balance : 800
 Exception in thread "main" NotSufficientFundException: Current balance 800 is less than requested
 amount 1000
 at Account.withdraw(Testing.java:68)
 at Testing.main(Testing.java:51)

EXPERIMENT-4.3

OBJECTIVE:

To create User Interface to perform Integer Divisions. The user enters two numbers in text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the result field when the divide button is clicked. If Num1 or Num2 were not integers, the program would throw a NumberFormatException. If Num2 is Zero, and the program would throw an ArithmeticException. Display the Exception in a message box.

RESOURCES:

Net Beans 8.0.2, JDK 7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name, A extends JFrame.
3. Declare the JLabel, JTextField, JButton variables.
4. Declare the default constructor of the class.
5. Add buttons, TextField and labels to the Flow Layout. 6. Call actionPerformed method.

SOURCE CODE:

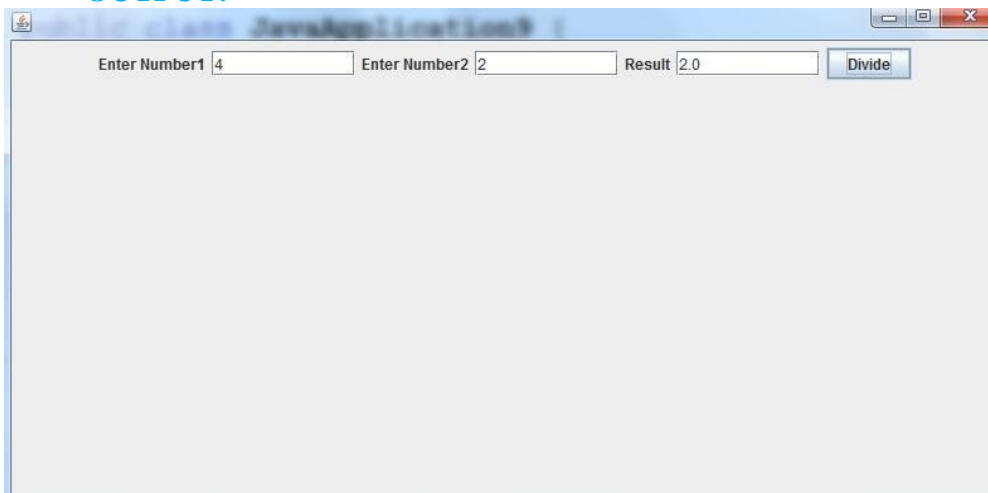
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
class A extends JFrame implements ActionListener {
    JLabel l1, l2, l3;
    JTextField tf1, tf2, tf3;
    JButton b1;
    A() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());
        l1 = new JLabel("Welcome");
        setSize(800, 400);
        l1 = new JLabel("Enter Number1");
        add(l1);
        tf1 = new JTextField(10);
        add(tf1);
        l2 = new JLabel("Enter Number2");
        add(l2);
        tf2 = new JTextField(10);
        add(tf2);
        l3 = new JLabel("Result");
        add(l3);
        tf3 = new JTextField(10);
```

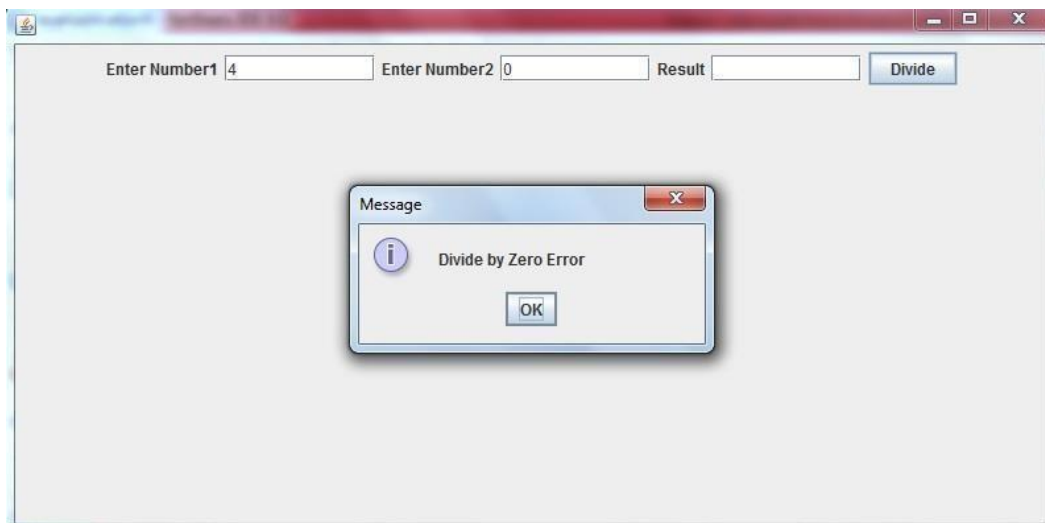
```

add(tf3);
b1 = new JButton("Divide");
add(b1);
b1.addActionListener(this);
setVisible(true);
}
public void actionPerformed(ActionEvent ae) {
try {
    int a = Integer.parseInt(tf1.getText());
    int b = Integer.parseInt(tf2.getText());
    if(b==0)
        throw new ArithmeticException(" Divide by Zero Error");
    float c = (float) a / b;
    tf3.setText(String.valueOf(c));
} catch (NumberFormatException ex) {
    JOptionPane.showMessageDialog(this, ex.getMessage());
} catch (ArithmeticException ex) {
    JOptionPane.showMessageDialog(this, ex.getMessage());
}
}
}
}
public class JavaApplication9
{
    public static void main(String[] args)
    {
        A a = new A();
    }
}

```

OUTPUT:





PRE LAB VIVA QUESTIONS:

1. Exceptions are defined in which java package?
2. How are the exceptions handled in java?
3. What is Runtime Exception or unchecked exception?
4. What is the difference between exception and error class?

POST LAB VIVA QUESTIONS:

5. What is checked exception?
6. What is an event and what are the models available for event handling?
7. What is difference between ClassNotFoundException and NoClassDefFoundError?
8. What is use of throws keyword?

WEEK – 5 MULTITHREADING

OBJECTIVE:

Write a java program that implements Thread class methods.

RESOURCES:

Net Beans 8.0.2, JDK 7.0.is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name „TestMethod1 extends Thread“.
3. Create thread objects and call thread class methods with thread object.
4. Pass the objects of our class to thread class .
5. Call the start method.

SOURCE CODE:

```
class TestMethod1 extends Thread
{
    public void run()
    {
        for(int i=1;i<=5;i++)
        {
            try
            {
                System.out.println("running thread name is:"+Thread.currentThread().getName());
                System.out.println("running thread priority is:"+Thread.currentThread().getPriority());
                Thread.sleep(500);
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
            System.out.println(i);
        }
    }
    public static void main(String args[]){
        TestMethod1 t1=new TestMethod1();
        TestMethod1 t2=new TestMethod1();
        TestMethod1 t3=new TestMethod1();
        t1.setPriority(Thread.MIN_PRIORITY);
        t2.setPriority(Thread.MAX_PRIORITY);
        t1.start();

        try{
            t1.join();

        }catch(Exception e){System.out.println(e);}
```

```

t2.start();
t3.start();
t1.setName("Sonoo");
    System.out.println("After changing name of t1:"+t1.getName());
}
}

```

OUTPUT:

running thread name is:Thread-0
running thread priority is:10
running thread name is:Thread-1
running thread priority is:1

Name of t1: Thread-0
Name of t2: Thread-1
id of t1:8
running...
After changeling name of t1: Sonoo

- 1
- 2
- 3
- 4
- 5
- 1
- 1
- 2
- 2
- 3
- 3
- 4
- 4
- 5
- 5

PRE LAB VIVA QUESTIONS:

- 1 .Define thread class?
2. Define runnable interface?
3. What is start method do?
4. What is run method??

POST LAB VIVA QUESTIONS:

1. What are thread states?
2. What are thread methods?
3. What is alive method?
4. What is thread interrupt?

EXPERIMENT-5.1

OBJECTIVE:

To implements a multithread application that has three threads. First thread generates random integer for every second, and if the value is even, second thread computes the square of number and prints .If the value is odd, the third thread will print the value of cube of number.

RESOURCES:

Net Beans 8.0.2, JDK 7.0.is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name „even implements Runnable“ and „odd implements Runnable“.
- 3.Create thread objects and Random class object. 4.Pass the objects of our class to thread class .
5. Call the start method.

SOURCE CODE:

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

class even implements Runnable {
public int x;
public even(int x) {
this.x = x;
}
public void run() {
System.out.println("Thread Name:Even Thread and " + x + "is even Number and Square of " + x + " is: "
+ x * x);
}
}

class odd implements Runnable {
public int x;
public odd(int x) {
this.x = x;
}
public void run() {
System.out.println("Thread Name:ODD Thread and " + x + " is odd number and Cube of " + x + " is: " +
x * x * x);
}
}

class A extends Thread {
public String tname;
public Random r;
```

```

public Thread t1, t2;
public A(String s) {
    tname = s;
}

public void run()
{
    int num = 0;
    r = new Random();
    try {
        for (int i = 0; i < 5; i++) {
            num = r.nextInt(100);
            System.out.println("Main Thread and Generated Number is " + num);
            if (num % 2 == 0) {
                t1 = new Thread(new even(num));
                t1.start();
            } else {
                t2 = new Thread(new odd(num));
                t2.start();
            }
            Thread.sleep(1000);
            System.out.println("-----");
        }
    }
    catch (Exception ex)
    {
        System.out.println(ex.getMessage());
    }
}

public class Mthread
{
    public static void main(String[] args)
    {
        A a = new A("One");
        a.start();
    }
}

```

OUTPUT:

Output - mthread (run)

```
run:
Main Thread and Generated Number is 95
Thread Name:ODD Thread and 95 is odd number and Cube of 95 is: 857375
-----
Main Thread and Generated Number is 90
Thread Name:Even Thread and 90is even Number and Square of 90 is: 8100
-----
Main Thread and Generated Number is 35
Thread Name:ODD Thread and 35 is odd number and Cube of 35 is: 42875
-----
Main Thread and Generated Number is 41
Thread Name:ODD Thread and 41 is odd number and Cube of 41 is: 68921
-----
Main Thread and Generated Number is 75
Thread Name:ODD Thread and 75 is odd number and Cube of 75 is: 421875
-----
Main Thread and Generated Number is 29
Thread Name:ODD Thread and 29 is odd number and Cube of 29 is: 24389
-----
Main Thread and Generated Number is 55
Thread Name:ODD Thread and 55 is odd number and Cube of 55 is: 166375
-----
Main Thread and Generated Number is 84
Thread Name:Even Thread and 84is even Number and Square of 84 is: 7056
-----
Main Thread and Generated Number is 36
Thread Name:Even Thread and 36is even Number and Square of 36 is: 1296
-----
Main Thread and Generated Number is 51
Thread Name:ODD Thread and 51 is odd number and Cube of 51 is: 132651
-----
Main Thread and Generated Number is 15
Thread Name:ODD Thread and 15 is odd number and Cube of 15 is: 3375
-----
```

PRE LAB VIVA QUESTIONS:

1. Why did the designers decide to force a method to specify all uncaught checked exceptions that can be thrown within its scope?
2. What are the states associated in the thread?
3. When you will synchronize a piece of your code?
4. What is the difference between process and thread?

POST LAB VIVA QUESTIONS:

5. What is multithreading and what are the methods for inter-thread communication and what is the class in which these methods are defined?
6. When you will synchronize a piece of your code?
7. What are daemon thread and which method is used to create the daemon thread?
8. What is deadlock?

Experiment 5.2

OBJECTIVE:

Write a java program that correct implements of producer consumer program.

RESOURCES:

Net Beans 8.0.2, JDK 7.0 is required, 1GB RAM, Hard Disk 80 GB.

SOURCE CODE:

```
class Q
{
    int n;
    boolean valueSet=false;
    synchronized int get()
    {
        if(!valueSet)
            try
            {
                wait();
            }
            catch(InterruptedException e)
            {
                System.out.println("InterruptedException caught");
            }
        System.out.println("Got:"+n);
        valueSet=false;
        notify();
        return n;
    }
    synchronized void put(int n)
    {
        if(valueSet)
            try
            {
                wait();
            }
            catch(InterruptedException e)
            {
                System.out.println("InterruptedException caught");
            }
        this.n=n;
        valueSet=true;
        System.out.println("Put:"+n);
        notify();
    }
}
```

```

class Producer implements Runnable
{
    Q q;
    Producer(Q q)
    {
        this.q=q;
        new Thread(this,"Producer").start();
    }
    public void run()
    {
        int i=0;
        while(true)
        {
            q.put(i++);
        }
    }
}
class Consumer implements Runnable
{
    Q q;
    Consumer(Q q)
    {
        this.q=q;
        new Thread(this,"Consumer").start();
    }
    public void run()
    {
        while(true)
        {
            q.get();
        }
    }
}
class ProdCons
{
    public static void main(String[] args)
    {
        Q q=new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-c to stop");
    }
}

```

OUTPUT:

Put:1 Got:1
Put:2 Got:2
Put:3 Got:3
Put:4 Got:4
Put:5 Got:5
Put:6 Got:6
Put:7 Got:7
Put:8 Got:8
Put:9 Got:9

PRE LAB VIVA QUESTIONS:

1. What is PC Problem?
2. What is Synchronization of threads?
3. What happens multiple objects acting on same object?
4. What is the difference between runnable and thread?

POST LAB VIVA QUESTIONS:

5. What is multithreading and multiprocessing?
6. When we use multithreading?
7. Difference between daemon thread and normal thread?
8. How to set thread priority?

WEEK-6: FILES

Experiment 6.1

OBJECTIVE

Write a Java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.

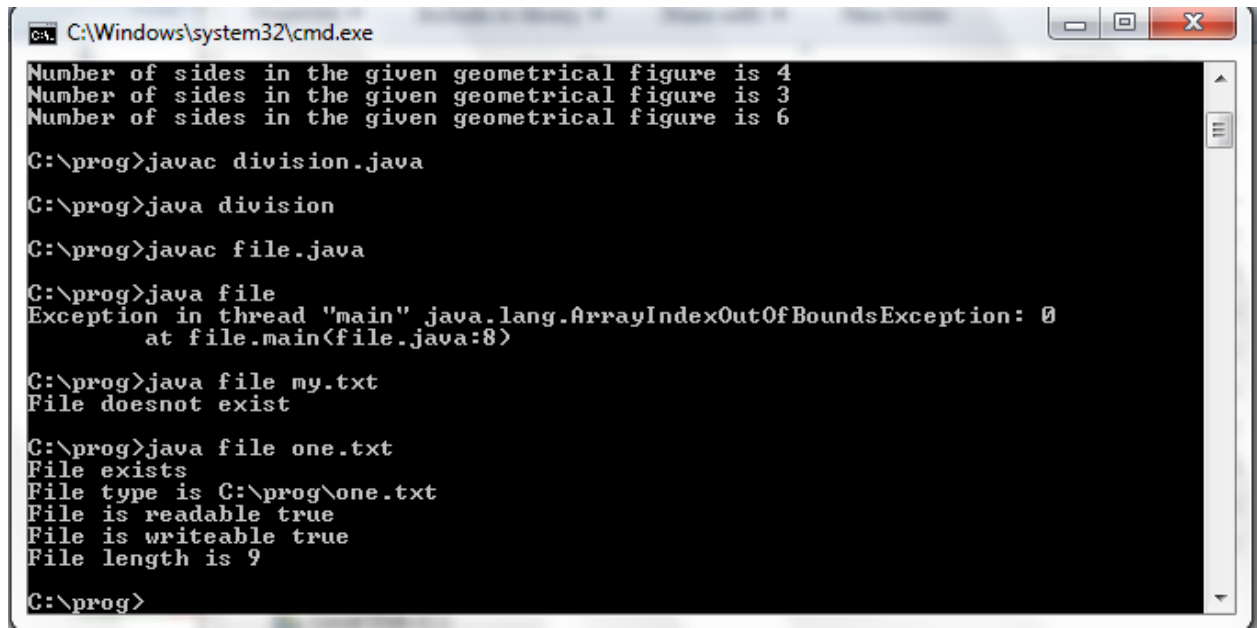
RESOURCES:

Net Beans 8.0.2, JDK7.0 is required, 1GB RAM, Hard Disk 80 GB

SOURCE CODE

```
import java.util.*;
import java.io.*;
import java.io.File;
class file
{
    public static void main(String args[])throws IOException
    {
        String s=args[0];
        File f=new File(s);
        if(f.exists())
        {
            System.out.println("File exists");
            System.out.println("File type is "+f.getAbsolutePath());
            System.out.println("File is readable "+f.canRead( ));
            System.out.println("File is writeable "+f.canWrite());
            System.out.println("File length is "+f.length());
        }
        else
        {
            System.out.println("File doesnot exist");
        }
    }
}
```

6.1.3 OUTPUT:



```
C:\Windows\system32\cmd.exe
Number of sides in the given geometrical figure is 4
Number of sides in the given geometrical figure is 3
Number of sides in the given geometrical figure is 6
C:\prog>javac division.java
C:\prog>java division
C:\prog>javac file.java
C:\prog>java file
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0
    at file.main(file.java:8)
C:\prog>java file my.txt
File doesnot exist
C:\prog>java file one.txt
File exists
File type is C:\prog\one.txt
File is readable true
File is writeable true
File length is 9
C:\prog>
```

PRE LAB VIVA QUESTIONS:

1. What is file?
2. How to create file input stream?
3. What is character stream?
4. What is byte stream?

POST LAB VIVA QUESTIONS:

1. How to convert the string data into long data?
2. Which input stream is used for primitive data?
3. What is reader class?
4. Define file attributes and its methods?

Experiment 6.2

OBJECTIVE

Write a Java program that displays the number of characters, lines and words in a text file.

RESOURCES:

Net Beans 8.0.2, JDK7.0 is required, 1GB RAM, Hard Disk 80 GB

SOURCE CODE

```
import java.io.*;
class wordcount
{
    public static int words=0;
    public static int lines=0;
    public static int chars=0;
    public static void wc(InputStreamReader isr)throws IOException
    {
        int c=0;
        boolean lastwhite=true;
        while((c=isr.read())!=-1)
        {
            chars++;
            if(c=='\n')
                lines++;
            if(c=='\t' || c==' ' || c=='\n')
                ++words;
            if(chars!=0)
                ++chars;
        }
    }
    public static void main(String[] args)
    {
        FileReader fr;
        try
        {
            if(args.length==0)
            {
                wc(new InputStreamReader(System.in));
            }
            else
            {
                for(int i=0;i<args.length;i++)
                {
                    fr=new FileReader(args[i]);
                    wc(fr);
                }
            }
        }
    }
}
```

```
        }  
    }  
    catch(IOException ie)  
    {  
        return;  
    }  
    System.out.println(lines+" "+words+" "+chars);  
}  
}
```

OUT PUT:

This is II CSE
1 4 32

Draw the frequency response
1 4 58

PRE LAB VIVA QUESTIONS:

1. What is File Input and output stream?
2. How to extract tokens?
3. What is a string class?
4. What is stringbuffer class?

POST LAB VIVA QUESTIONS:

1. How to convert the string data into long data?
2. Which containers use a Border layout as their default layout?
3. What is an event and what are the models available for event handling?
4. Which listeners are implemented for JButton ?

Experiment 6.3

OBJECTIVE:

Write a Java program that reads a file and displays the file on the screen, with a line number before each line.

RESOURCES:

Net Beans 8.0.2, JDK7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Create file and store some information.
2. Create class linenum.
3. Read that file created already display line number before each line.

SOURCE CODE

```
import java.io.*;

class linenum
{
    public static void main(String[] args) throws IOException
    {
        FileInputStream fil;
        LineNumberInputStream line;
        int i;
        try
        {
            fil=new FileInputStream(args[0]);
            line=new LineNumberInputStream(fil);
        }
        catch(FileNotFoundException e)
        {
            System.out.println("No such file found");
            return;
        }
        do
        {
            i=line.read();
            if(i=='\n')
            {
                System.out.println();
                System.out.print(line.getLineNumber()+" ");
            }
            else
                System.out.print((char)i);
        } while(i!=-1);
        fil.close();
        line.close();
    }
}
```

```
}
```

OUT PUT:

Demo.java

```
class Demo
1 {
2 public static void main(java Demo beta gamma delta)
3 {
4 int n = 1 ;
5 System.out.println("The word is " + args[ n ]);
6 }
6 }
8?
```

PRE LAB VIVA QUESTIONS

1. What is file class?
2. What are common methods in streams?
3. What is the difference between reader class and writer class?
4. What is piped input stream?

POST LAB VIVA QUESTIONS

1. What are input stream classes?
2. What are output stream classes?
3. What is buffered reader class?
4. How to file object?

WEEK-7: FILES

EXPERIMENT-7.1

OBJECTIVE:

To display a table using Label in Grid Layout. Suppose that a table named Table.txt is stored in a text file. The First line in the file is the header, and the remaining lines correspond rows in table. The elements are separated by commas Write a java program to display the table using labels in Grid layout.

RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name A implements ItemListener.
3. Create the FileInputStream object.
4. Create the Scanner object delimited by comma.
5. Close the JFrame.

SOURCE CODE:

```
import java.io.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

class A extends JFrame
{
    public A()
    {
        setSize(400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        GridLayout g = new GridLayout(0, 3);
        setLayout(g);
    }
}
```

```

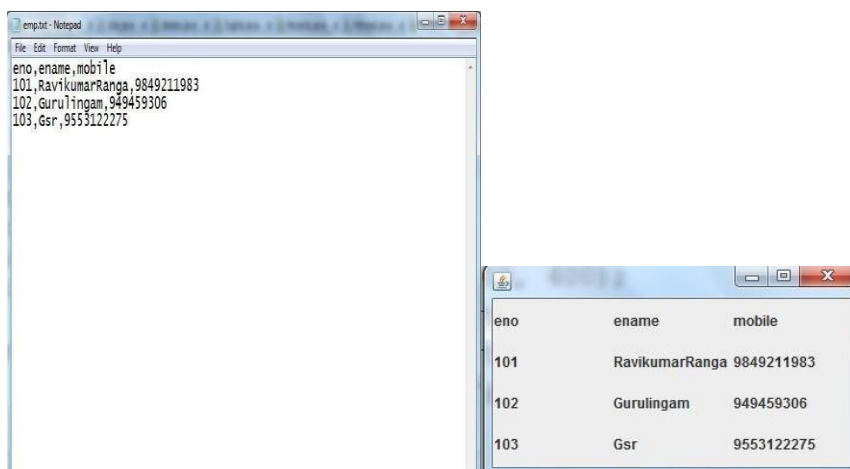
try
{
    FileInputStream fin = new FileInputStream("D:\\table.txt");

    Scanner sc = new Scanner(fin).useDelimiter(",");
    String[] arrayList;
    String a;
    while (sc.hasNextLine())
    {
        a = sc.nextLine();
        arrayList = a.split(",");
        for (String i : arrayList)
        {
            add(new JLabel(i));
        }
    }
} catch (Exception ex) {
}
setDefaultLookAndFeelDecorated(true);
pack();
setVisible(true);
}
}

public class Tabledemo {
public static void main(String[] args) {
A a = new A();
}
}
Emp.txt

```

INPUT AND OUTPUT:



PRE LAB VIVA QUESTIONS

1. What is the difference between gui and cui?
2. What is awt?
3. How to enable the frame ?
4. Which container is best suitable for gui type of applications?

POST LAB VIVA QUESTIONS

5. What are different types of layout manager in awt?
6. What is difference between jframe and swing?
7. What are different types of components present?
8. What is the root class for all container classes?

EXPERIMENT-7.2

OBJECTIVE:

Write a java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.

RESOURCES:

Net Beans 8.0.2, JDK7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name „CRUD“.
3. Load the Driver class
4. Create a Connection object
5. Create a Statement object
6. Get the Connection and execute the create, retrieve, update and delete operations on an SQL Command
7. Close the Connection object, Statement object. st.close();
cn.close();

SOURCE CODE:

```
import java.sql.*;
import javax.sql.*;
import java.util.*;
public class CRUD
{
public static void main(String[] args)
    {
        Connection cn;
        Statement st;
        ResultSet rs;
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/IARE", "root", "98765432");
            st = cn.createStatement();
            System.out.println("Welcome To Institute of Aeronautical Engineering College");
            System.out.println("-----MENU ----- ");
            System.out.println("1.Insert");
            System.out.println("2.EDIT");
            System.out.println("3.Delete");
            System.out.println("4.Display");
            System.out.println("5.Exit");
        }
    }
}
```

```

System.out.println("----- ");
String opt = "";
String htno = "", sname = "", mobile = "", sql = "";
Scanner sc = new Scanner(System.in);
while (opt != "5")
{
System.out.println("Enter Your Option");
opt = sc.next();
switch (opt)
{
case "1":
{
System.out.println("Enter Htno");
htno = sc.next();
System.out.println("Enter Name");
sname = sc.next();
System.out.println("Enter Mobile");
mobile = sc.next();
sql = "insert into stu values(" + "" + htno + "" + "," + "" + sname + "" + "," + "" +
mobile + "" + ")";
if (st.executeUpdate(sql) > 0)
{
System.out.println("Record Inserted");
}
}
}
break;
case "2":
{
System.out.println("Enter Htno to Update");
htno = sc.next();
System.out.println("Enter Name");
sname = sc.next();
System.out.println("Enter Mobile");
mobile = sc.next();
sql = "update stu set sname=" + "" + sname + "" + "," + "mobile=" + "" + mobile +
"" + " where htno=" + htno + """;
if (st.executeUpdate(sql) > 0) {
System.out.println("Record Updated");
}
else
{
System.out.println("Operation Failed");
}
}
}
break;
case "3":
{
System.out.println("Enter Htno to delete");
htno = sc.next();
sql = "delete from stu where Htno=" + "" + htno + """;
}
}
}

```

```

        if (st.executeUpdate(sql) > 0) {
            System.out.println("Record deleted");
        } else {
            System.out.println("Operation Failed");
        }
    }
    break;
    case "4": {
        sql = "select * from stu";
        rs = st.executeQuery(sql);
        System.out.println("Htno\tSname\tMobile");
        while (rs.next()) {
            System.out.println(rs.getString("Htno") + "\t" + rs.getString("SName") + "\t" +
                rs.getString("mobile"));
        }
        rs.close();
    }
    break;
    case "5":
    {
        opt = "5";
        System.out.println("Thank You");
        st.close();
        cn.close();
    }
    break;
    default: {
        System.out.println("Choose Option between 1 and 5 only"); }
    }
}
}
catch (Exception ex)
{
    System.out.println(ex.getMessage());
}
}
}
}

```


OUTPUT:

Output - CRUD (run)

```
run:
Welcome To Institute of Aeronautical Engineering College
-----MENU-----
1.Insert
2.EDIT
3.Delete
4.Display
5.Exit
-----
Enter Your Option
|
```

Output - CRUD (run)

```
-----MENU-----
1.Insert
2.EDIT
3.Delete
4.Display
5.Exit
-----
Enter Your Option
1
Enter Htno
101
Enter Name
RavikumarRanga
Enter Mobile
9849211983
Record Inserted
Enter Your Option
4
Htno      Sname      Mobile
101      RavikumarRanga  9849211983
Enter Your Option
2
Enter Htno to Update
101
Enter Name
RavikumarRanga
Enter Mobile
9849211983
Record Updated
Enter Your Option
3
Enter Htno to delete
101
Record deleted
```

PRE LAB VIVA QUESTIONS

1. What is JDBC?
2. What are drivers available?
3. What is the difference between JDBC and ODBC?
4. What is stored procedure?

POST LAB VIVA QUESTIONS

1. What are the types of JDBC Driver Models and explain them?
2. What are the steps involved for making a connection with a database or how do you connect to a database?
3. What are the types of statements in JDBC?
4. How to create and call stored procedures?

WEEK-8: JAVA PROGRAM WITH DATABASE

EXPERIMENT-8.1

OBJECTIVE:

Write a java program that loads names and phone numbers from the text file where data is organized as one line per record and each field in record are separated by a tab(\t). It takes a name or phone number as input and prints corresponding other value from hash table(hint: use Hash Table).

RESOURCES:

Net Beans 8.0.2, JDK7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Create a text file which contains names and phone numbers separated by a tab
2. Using buffer reader the file has to be read and the content has to be moved to hash table using put method.
3. Whenever a name is given the corresponding phone number from hash table has to be displayed

SOURCE CODE:

```
import java.util.*;
import java.io.*;
public class PhoneDictionary
{
    public static void main(String[] args)
    {
        try
        {
            FileInputStream fs = new FileInputStream("D:\\ph.txt");
            Scanner sc = new Scanner(fs).useDelimiter("\\s+");
            Hashtable<String, String> ht = new Hashtable<String, String>();
            String[] arrayList;
            String a;
            System.out.println("Welcome to Institute of Aeronautical Engineering");
            System.out.println("Student Phone numbers are");
            while (sc.hasNext())
            {
```

```

        a = sc.nextLine();
        arrayList = a.split("\\s+");
        ht.put(arrayList[0], arrayList[1]);
        System.out.println(arrayList[0] + ":" + arrayList[1]);
    }

    System.out.println("iare");
    System.out.println("MENU");
    System.out.println("1.Search by Name");
    System.out.println("2.Search by Mobile");
    System.out.println("3.Exit");
    String opt = "";
    String name, mobile;
    Scanner s = new Scanner (System.in);
    while (opt != "3")
    {
        System.out.println("Enter Your Option (1,2,3): ");
        opt = s.next();
        switch (opt)
        {
            case "1":
            {
                System.out.println("Enter Name");
                name = s.next();
                if (ht.containsKey(name))
                {
                    System.out.println("Mobile is " + ht.get(name));
                }
                else
                {
                    System.out.println("Not Found");
                }
            }
            break;
            case "2":
            {
                System.out.println("Enter mobile");
                mobile = s.next();
                if (ht.containsValue(mobile))
                {
                    for (Map.Entry e : ht.entrySet())
                    {
                        if (mobile.equals(e.getValue()))
                        {
                            System.out.println("Name is " + e.getKey());
                        }
                    }
                }
            }
            else
            {

```

```
        System.out.println("Not Found");
    }
}

break;
case "3":
{
opt = "3";
System.out.println("Menu Successfully Exited");
}
break;
default:
System.out.println("Choose Option between 1 and Three");
break;

}
}
}

catch (Exception ex)

{
System.out.println(ex.getMessage());
}
}
}
```

INPUT:

Loads names and phone numbers from the text file named ph.txt

OUPUT:

```
D:\JP>java PhoneDictionary
iare
Student Phone numbers are
P.RaghuNandan:8977332085
P.Ronith:9052277577
Y.Srinish:9849039239
Y.Snikitha:9849098490
P.Rishikesh:9848098480
Iare
MENU
1.Search by Name
2.Search by Mobile
3.Exit
Enter Your Option( 1,2,3):
1
Iare
22
Enter Name
P.RaghuNandan
Mobile is 8977332085
Enter Your Option( 1,2,3):
2
Enter mobile
9849098490
Name is Y.Snikitha
Enter Your Option 1,2,3
3
Menu Successfully Exited
```

PRE LAB VIVA QUESTIONS

1. What is JDBC and its drivers?
2. What are prepared statements?
3. What is ODBC?
4. What is driver manager?

POST LAB VIVA QUESTIONS

1. What are the types of JDBC Driver?
2. What are the steps involved for making a connection with a database or how do you connect to a database?
3. What are the types of statements in JDBC?
4. How to use dml commands?

EXPERIMENT-8.2

OBJECTIVE:

Implement above program with database instead of text file.

RESOURCES:

Net Beans 8.0.2, JDK7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Create a text file which contains names and phone numbers separated by a tab
2. Using buffer reader the file has to be read and the content has to be moved to hash table using put method.
3. Whenever a name is given the corresponding phone number from hash table has to be displayed

SOURCE CODE:

```
import java.sql.*;
import java.util.*;
public class PhoneDB
{
    public static void main(String[] args)
    {
        Connection cn;
        Statement st;
        ResultSet rs;
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            cn = DriverManager.getConnection("jdbc:odbc:siet","siet","siet");
            System.out.println("Data Base Connected");
            st = cn.createStatement();
            rs = st.executeQuery("select * from phone");
            Hashtable<String, String> ht = new Hashtable<String, String>();
            System.out.println("IARE");
            System.out.println("Student Phone numbers are");
            while (rs.next())
            {
                ht.put(rs.getString(1), rs.getString(2));
                System.out.println(rs.getString(1) + ":" + rs.getString(2));
            }
            rs.close();
            st.close();
            cn.close();
            System.out.println("-----");
            System.out.println("Welcome IARE");
            System.out.println("Main Menu");
            System.out.println("1.Search by Name");
            System.out.println("2.Search by Mobile");
        }
    }
}
```

```

System.out.println("3.Exit");
String opt = "";
String name, mobile;
Scanner s = new Scanner(System.in);
while (opt != "3")
{
System.out.println("-----");
System.out.println("Enter Your Option 1,2,3");
opt = s.next();
switch (opt)
{
case "1":
System.out.println("Enter Name");
name = s.next();
if (ht.containsKey(name))
{
System.out.println("Mobile is " + ht.get(name));
} else {
System.out.println("Not Found");
}
break;
case "2":
System.out.println("Enter mobile");
mobile = s.next();
if (ht.containsValue(mobile))
{
for (Map.Entry e : ht.entrySet())
{
if (mobile.equals(e.getValue()))
{
System.out.println("Name is " + e.getKey());
}
}
} else {
System.out.println("Not Found");
}
break;
case "3":
opt = "3";
System.out.println("Menu Successfully Exited");
break;
default:
System.out.println("Choose Option between 1 and Three");
}
} catch (Exception ex) {
System.out.println(ex.getMessage());
}
}
}

```


INPUT from database Table in Oracle:

```
1) Create a table name as "phone"
create table phone(stname varchar2(20), stmobile varchar2(20));
2) Insert records into DataBase Table "phone"
insert into phone values('P.Raghu','8977332085');
insert into phone values('Ramu','9849098490');
insert into phone values('Hari','9877335566');
insert into phone values('Jayadev','9052778877');
insert into phone values('Harika','9849044556');
3) Retrieve the table
select * from phone;
Table displays as
```

OUTPUT:

```
D:\JP>javac PhoneDB.java
D:\JP>java PhoneDB
Data Base Connected
Student Phone numbers are
P.Raghu:8977332085
Ramu:9849098490
Hari:9877335566
Jayadev:9052778877
Harika:9849044556
```

Main Menu

```
1.Search by Name
2.Search by Mobile
3.Exit
```

Enter Your Option 1,2,3

1

Enter Name

Jayadev

Mobile is 9052778877

Enter Your Option 1,2,3

2

Enter mobile

8977332085

Name is P.Raghu

Enter Your Option 1,2,3

3

Menu Successfully Exited

PRE LAB VIVA QUESTIONS:

1. What are the JDBC API components?
2. What are the JDBC statements?
3. What is the difference between Statement and PreparedStatement interface?
4. How can we execute stored procedures and functions?

POST LAB VIVA QUESTIONS:

5. What is the role of JDBC DriverManager class?
6. What does the JDBC Connection interface?
7. What does the JDBC ResultSet interface?
8. What does the JDBC ResultSetMetaData interface?

WEEK-9: FILES

EXPERIMENT: 9.1

OBJECTIVE:

Write a Java Program that takes tab separated data (one record line) from a text file and insert them into a database.

RESOURCES:

Net Beans 8.0.2, JDK7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Create a text file which contains names and phone numbers separated by a tab
2. Using buffer reader the file has to be read and the content has to be moved to hash table using put method.
3. Whenever a name is given the corresponding phone number from hash table has to be displayed

SOURCE CODE:

```
import java.sql.*;
import java.io.*;
import java.util.*;
public class Tbltodb
{
    public static void main(String[] args)
    {
        Connection cn;
        Statement st;
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            cn=DriverManager.getConnection("jdbc:odbc:siet","siet","siet");
            st=cn.createStatement();
            String sql="";
            FileInputStream fin=new FileInputStream("D:\\emp.txt");
            Scanner sc=new Scanner(fin);
            String[] arrayList;
            String a="";
            int i=0;
            while(sc.hasNext())
            {
                a=sc.nextLine();
                arrayList =a.split("\\s+");
                sql="insert into employee
                values('"+arrayList[0]+"','"+arrayList[1]+"','"+arrayList[2]+"')";
                st.execute(sql);
                i++;
            }
        }
    }
}
```

```

        System.out.println(arrayList[0]+":"+arrayList[1]+":"+arrayList[2]);
    }
    System.out.println(i+" Records are inserted");
    st.close();
    cn.close();
}
catch(Exception ex)
{
    System.out.println(ex.getMessage());
}
}
}

```

OUTPUT:

```

D:\JP>javac Tbltodb.java
D:\JP>java Tbltodb
CSE1001:P.Raghu:8977332085
CSE1002:G.Uday:9849677889
CSE1003:Kavitha:8949088993
CSE1004:Swetha:7805805805
4 Records are inserted
After Inserted records in the Data Base Empllyee table as
Sql>select * from employee;
Institute of Aeronautical Engineering Java Programming Record

```

PRE LAB VIVA QUESTIONS:

1. What does the JDBC DatabaseMetaData interface?
2. Which interface is responsible for transaction management in JDBC?
3. What is batch processing and how to perform batch processing in JDBC?
4. How can we store and retrieve images from the database?

POST LAB VIVA QUESTIONS:

1. What is CallableStatement?
2. What is Statement?
3. What is PreparedStatement?
4. What is ResultSet?

EXPERIMENT: 9.2

OBJECTIVE:

Write a Java Program that prints the meta data of a given table.

RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name Tblmdata.
3. Load the Driver class
4. Create a Connection object
5. Create a Statement object
6. Get the Connection and execute the create, retrieve, update and delete operations on an SQL Command
7. Close the Connection object, Statement object.
st.close();
cn.close();

PROCEDURE:

To execute a java program we require setting a class path:

- 1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
- 2.C:\javac Tblmdata.java
- 3.C:\java Tblmdata

SOURCE CODE:

```
import java.sql.*;
import java.util.*;
public class Tblmdata {
public static void main(String[] args) {
Connection cn;
Statement st;
ResultSet rs, rs1;
ResultSetMetaData rsmd;
try {
Scanner sc = new Scanner(System.in);
System.out.println("-----Welcome to Institute of Aeronautical Engineering College -----");
System.out.println("Enter Database Name");
String dbname = sc.next();
System.out.println("Enter Password");
String pass = sc.next();
Class.forName("com.mysql.jdbc.Driver");
```

```

cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/" + dbname, "root", pass);
st = cn.createStatement();
DatabaseMetaData dm = cn.getMetaData();
rs = dm.getTables(cn.getCatalog(), "%", "%", null);
String s = "";
String sql = "select * from ", sql1 = "";
System.out.println(" -----Database is " + dbname);
System.out.println(" -----");
System.out.println("Tables are");
System.out.println(" -----");
while (rs.next()) {
sql1 = "";
System.out.println("-----Table Name: " + rs.getString(3) + "-----");
sql1 = sql + rs.getString(3);
rs1 = st.executeQuery(sql1);
rsmd = rs1.getMetaData();
System.out.println("Columns are ");
System.out.println("Column Name\tColumn Type\tSize");
for (int i = 1; i <= rsmd.getColumnCount(); i++) {
System.out.println(rsmd.getColumnLabel(i) + "\t" + rsmd.getColumnTypeName(i) + "\t" +
rsmd.getColumnDisplaySize(i));
}
System.out.println(" -----");
}
rs.close();
cn.close();
} catch (Exception ex) {
System.out.println(ex.getMessage());
}
}
}

```

INPUT AND OUTPUT:

```
Output
Hashtbl2 (run)  ss  tblmdata (run)  ss

run:
Welcome to Institute of Aeronautical Engineering College
Enter Database Name
anurag
Enter Password
12345678
-----Database is anurag
-----
Tables are
-----
-----Table Name: emp-----
Columns are
Column Name      Column Type      Size
eno      VARCHAR 50
ename    VARCHAR 50
mobile   VARCHAR 50
-----
-----Table Name: phone-----
Columns are
Column Name      Column Type      Size
name     VARCHAR 50
mobile   VARCHAR 50
-----
-----Table Name: stu-----
Columns are
Column Name      Column Type      Size
Htno     VARCHAR 50
Sname     VARCHAR 50
mobile   VARCHAR 50
-----
-----Table Name: wp_commentmeta-----
Columns are
Column Name      Column Type      Size
meta_id BIGINT UNSIGNED 20
comment_id      BIGINT UNSIGNED 20
```

PRE LAB VIVA QUESTIONS

1. What is Java Data Base Connectivity?
2. How to create a table in a data base?
3. How to retrieve a table in the database?
4. What is meta data?

POST LAB VIVA QUESTIONS

1. What is meta data?
2. How to create a connection object?
3. What is result set object?
4. How do we get column count?

WEEK-10: TRAFIC LIGHT

EXPERIMENT-10.1

OBJECTIVE:

To simulate a Traffic Light. The program lets the user select one of three lights: red, yellow or Green with radio buttons. On selecting radio button, an appropriate message with “stop” or “Ready” or “GO” should appear above the button in selected color. Initially, there is no message Shown.

RESOURCES:

Net Beans 8.0.2, JDK7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name A implements ItemListener.
3. Create the ButtonGroup ,JRadiobuttons,JPanel.
4. Add to the JFrame
5. Register the components to the JFrame.
6. Close the JFrame.

SOURCE CODE:

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
class A extends JFrame implements ItemListener
{
    public JLabel l1, l2;
    public JRadioButton r1, r2, r3;
    public ButtonGroup bg;
    public JPanel p, p1;
    public A()
    {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(2, 1));
        setSize(800, 400);
        p = new JPanel(new FlowLayout());
        p1 = new JPanel(new FlowLayout());
        l1 = new JLabel();
        Font f = new Font("Verdana", Font.BOLD, 60);
        l1.setFont(f);
        add(l1);
        p.add(l1);
        add(p);
        l2 = new JLabel("Select Lights");
        p1.add(l2);
```



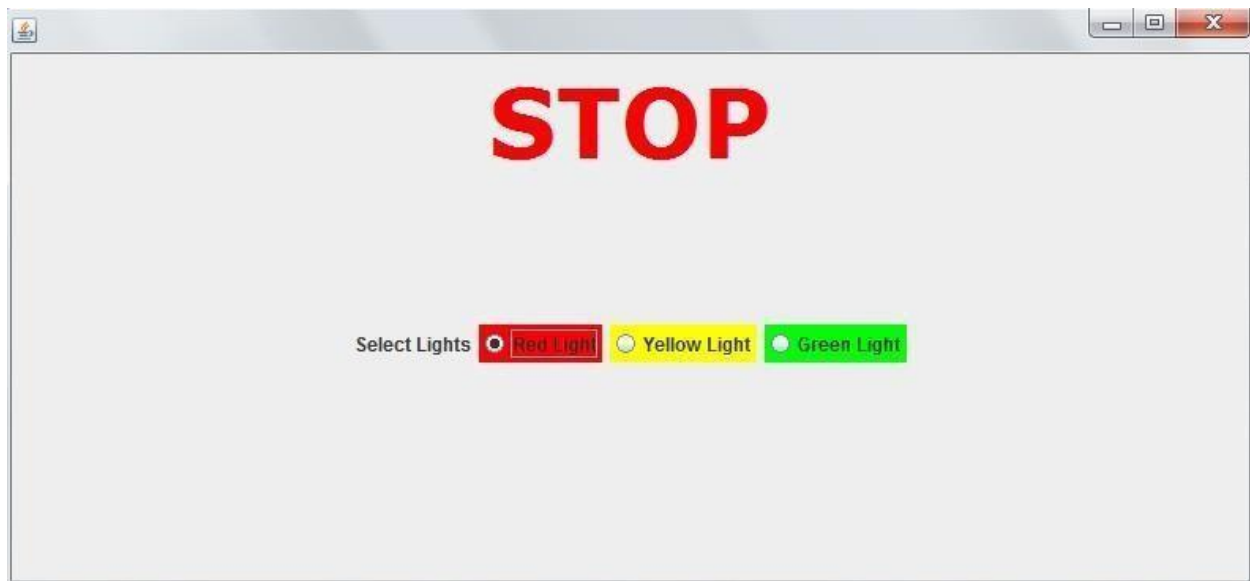
```

        JRadioButton r1 = new JRadioButton("Red Light");
        r1.setBackground(Color.red);
        p1.add(r1);
        r1.addItemListener(this);
        JRadioButton r2 = new JRadioButton("Yellow Light");
        r2.setBackground(Color.YELLOW);
        p1.add(r2);
        r2.addItemListener(this);
        JRadioButton r3 = new JRadioButton("Green Light");
        r3.setBackground(Color.GREEN);
        p1.add(r3);
        r3.addItemListener(this);
        add(p1);
        bg = new ButtonGroup();
        bg.add(r1);
        bg.add(r2);
        bg.add(r3);
        setVisible(true);
    }
    public void itemStateChanged(ItemEvent i)
    {
        JRadioButton jb = (JRadioButton) i.getSource();
        switch (jb.getText())
        {
            case "Red Light":
            {
                l1.setText("STOP");
                l1.setForeground(Color.RED);
            }
            break;
            case "Yellow Light":
            {
                l1.setText("Ready");
                l1.setForeground(Color.YELLOW);
            }
            break;
            case "Green Light":
            {
                l1.setText("GO");
                l1.setForeground(Color.GREEN);
            }
            break;
        }
    }
}

```

```
public class TLights
{
    public static void main(String[] args)
    {
        A a = new A();
    }
}
```

INPUT AND OUTPUT:





PRE LAB VIVA QUESTIONS:

1. What is a layout manager and what are different types of layout managers available in java AWT?
2. How are the elements of different layouts organized?
3. What is Garbage Collection and how to call it explicitly?
4. What is the difference between Integer and int?

POST LAB VIVA QUESTIONS:

5. What are Vector, Hashtable, LinkedList and Enumeration?
6. What is a better way of creating multithreaded application? Extending Thread class or implementing Runnable?
7. Can the start() method of the Thread class be overridden? If yes should it be overridden?
8. List the methods which when called on the object the thread releases the locks held on that object?

WEEK-11: MOUSE EVENTS

EXPERIMENT-11.1

OBJECTIVE:

To handle all mouse events and show event name at the center of the window when the mouse event is fired.(Use Adapter Classes)

RESOURCES:

Net Beans 8.0.2, JDK.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

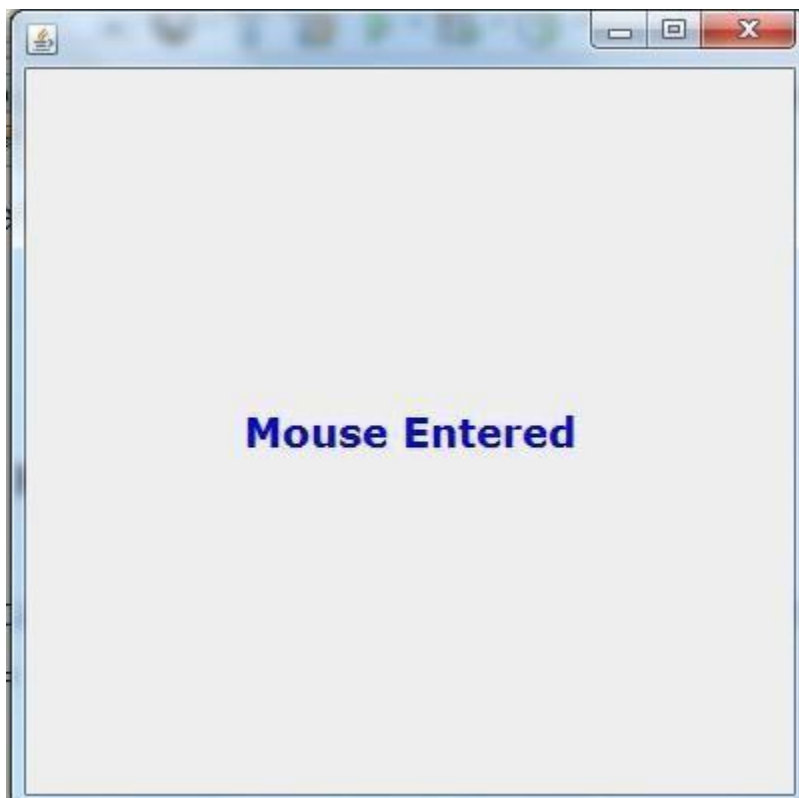
1. Start the program
2. Create a class with the name A implements MouseListener.
3. Call the mouse events.
4. Add to the JFrame
5. Register the components to the JFrame.
6. Close the JFrame.

SOURCE CODE:

```
import javax.swing.*;
import java.awt.*;
import javax.swing.event.*;
import java.awt.event.*;
class A extends JFrame implements MouseListener
{
    JLabel l1;
    public A()
    {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 400);
        setLayout(new GridBagLayout());
        l1 = new JLabel();
        Font f = new Font("Verdana", Font.BOLD, 20);
        l1.setFont(f);
        l1.setForeground(Color.BLUE);
        l1.setAlignmentX(Component.CENTER_ALIGNMENT);
        l1.setAlignmentY(Component.CENTER_ALIGNMENT);
        add(l1);
        addMouseListener(this);
        setVisible(true);
    }
    public void mouseExited(MouseEvent m)
    {
        l1.setText("Mouse Exited");
    }
}
```

```
}  
public void mouseEntered(MouseEvent m)  
{  
    l1.setText("Mouse Entered");  
}  
public void mouseReleased(MouseEvent m)  
{  
    l1.setText("Mouse Released");  
}  
public void mousePressed(MouseEvent m)  
{  
    l1.setText("Mouse Pressed");  
}  
public void mouseClicked(MouseEvent m)  
{  
    l1.setText("Mouse Clicked");  
}  
}  
}  
public class Mevents { public static void main(String[] args) {  
    A a = new A();  
}  
}
```

INPUT AND OUTPUT:





PRE LAB VIVA QUESTIONS:

1. What are different types of events?
2. How do we create mouse events?
3. What are different types of mouse events?
4. What are different types of objects created?

POST LAB VIVA QUESTIONS:

5. When the mouse pressed event is triggered?
6. When the mouse released event is triggered?
7. When the mouse entered event is triggered?
8. What are different types of listeners?

EXPERIMENT-11.2

OBJECTIVE:

Write a Java program to demonstrate the key event handlers.

RESOURCES:

Net Beans, JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name KeyListenerExample extends Frame implements KeyListener.
3. Call the mouse events.
4. Add to the JFrame
5. Register the components to the JFrame.
6. Close the JFrame.

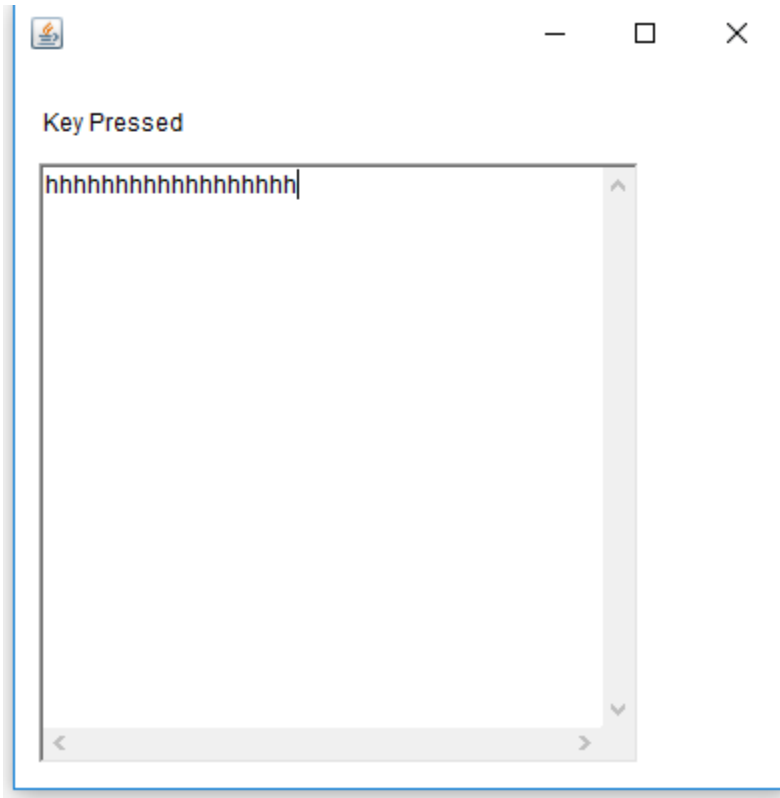
SOURCE CODE:

```
import java.awt.*;
import java.awt.event.*;
public class KeyListenerExample extends Frame implements KeyListener{
    Label l;
    TextArea area;
    KeyListenerExample(){

        l=new Label();
        l.setBounds(20,50,100,20);
        area=new TextArea();
        area.setBounds(20,80,300, 300);
        area.addKeyListener(this);

        add(l);add(area);
        setSize(400,400);
        setLayout(null);
        setVisible(true);
    }
    public void keyPressed(KeyEvent e) {
        l.setText("Key Pressed");
    }
    public void keyReleased(KeyEvent e) {
        l.setText("Key Released");
    }
    public void keyTyped(KeyEvent e) {
        l.setText("Key Typed");
    }
}
```

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



PRE LAB VIVA QUESTIONS:

1. Which containers use a border layout as their default layout?
2. Which containers use a FlowLayout as their default layout?
3. What is AWT?
4. What is the super class of all containers?

POST LAB VIVA QUESTIONS:

5. What is component?
6. How to interact with the Java system at runtime?
7. What is the super class of all components of Java?
8. What is a container?

WEEK-12: CALCULATOR

EXPERIMENT-12.1

OBJECTIVE:

To develop a program that works as simple calculator .Uses grid layout to arrange buttons for the digits and for the +,-,*, % operations. Add text field to display the results, Handle any possible exceptions like divide by zero.

RESOURCES:

Net Beans, JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name Cal that extendsApplet implements ActionListener.
3. Declare the Button ,TextField variables.
4. Add listener to each button and to the layout.
5. Add Text Field to display the result.
6. Handle any Exceptions like divide by zero.

SOURCE CODE:

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

/*
<applet code="Cal" width=300 height=300>
</applet>
*/

public class Cal extends Applet implements ActionListener
{
    String msg=" ";
    int v1,v2,result;
    TextField t1;
    Button b[]=new Button[10];
    Button add,sub,mul,div,clear,mod,EQ;
    char OP;
    public void init()
    {
        Color k=new Color(120,89,90);
        setBackground(k);
```

```

t1=new TextField(10);
GridLayout gl=new GridLayout(4,5);
setLayout(gl);
for(int i=0;i<10;i++)
{
    b[i]=new Button(""+i);
}
add=new Button("add");
sub=new Button("sub");
mul=new Button("mul");
div=new Button("div");
mod=new Button("mod");
clear=new Button("clear");
EQ=new Button("EQ");
t1.addActionListener(this);
add(t1);
for(int i=0;i<10;i++)
{
    add(b[i]);
}
add(add);
add(sub);
add(mul);
add(div);
add(mod);
add(clear);
add(EQ);
for(int i=0;i<10;i++)
{
    b[i].addActionListener(this);
}
add.addActionListener(this);
sub.addActionListener(this);
mul.addActionListener(this);
div.addActionListener(this);
mod.addActionListener(this);
clear.addActionListener(this);
EQ.addActionListener(this);
}

public void actionPerformed(ActionEvent ae)
{
    String str=ae.getActionCommand();
    char ch=str.charAt(0);
    if ( Character.isDigit(ch))

```

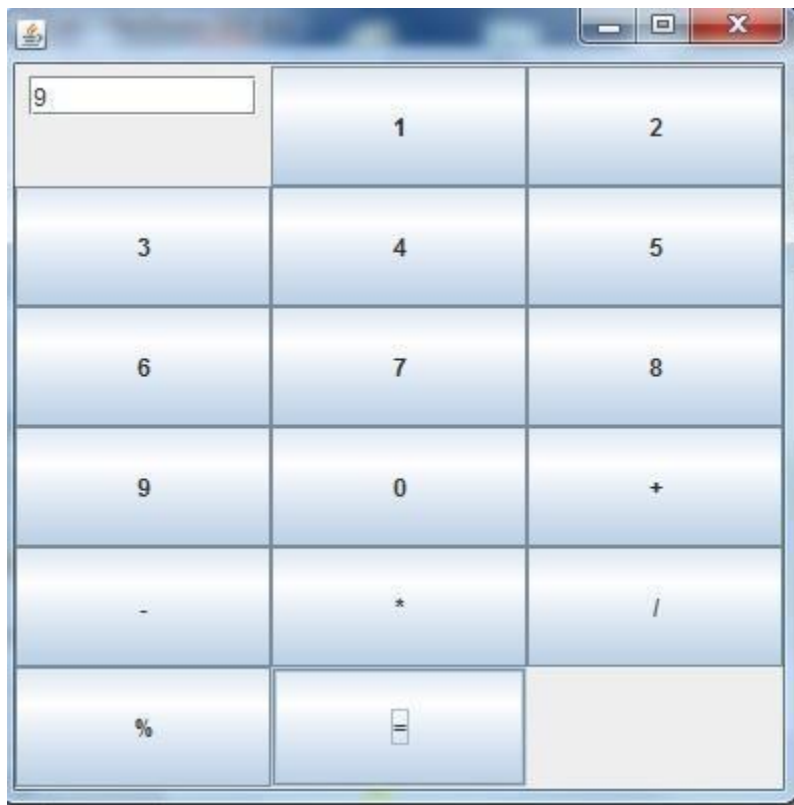
```

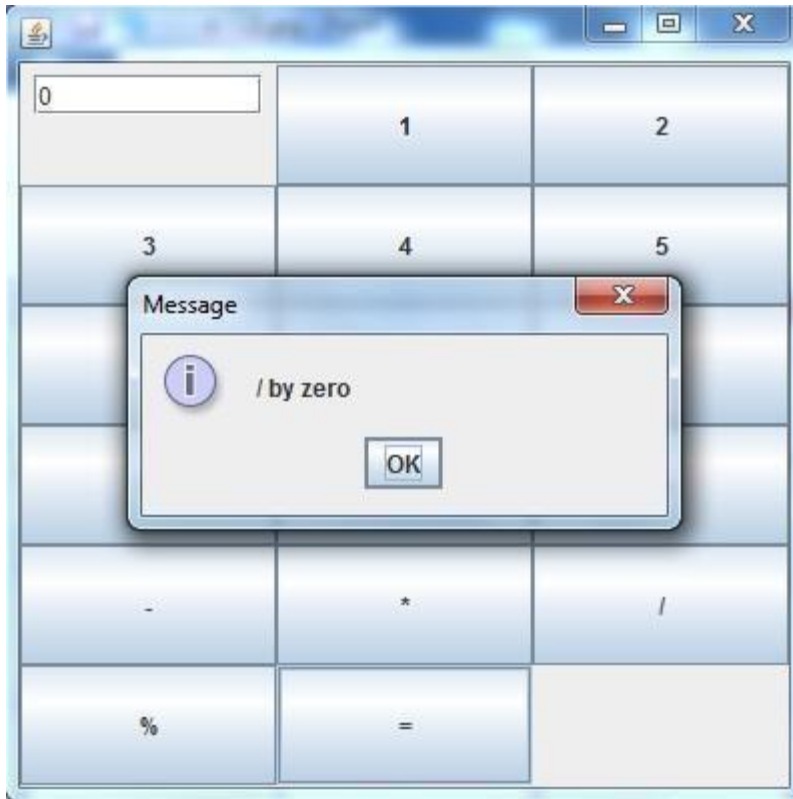
t1.setText(t1.getText()+str);
else
if(str.equals("add"))
{
    v1=Integer.parseInt(t1.getText());
    OP='+';
    t1.setText("");
}
else if(str.equals("sub"))
{
    v1=Integer.parseInt(t1.getText());
    OP='-';
    t1.setText("");
}
else if(str.equals("mul"))
{
    v1=Integer.parseInt(t1.getText());
    OP='*';
    t1.setText("");
}
else if(str.equals("div"))
{
    v1=Integer.parseInt(t1.getText());
    OP='/';
    t1.setText("");
}
else if(str.equals("mod"))
{
    v1=Integer.parseInt(t1.getText());
    OP='%';
    t1.setText("");
}
if(str.equals("EQ"))
{
    v2=Integer.parseInt(t1.getText());
    if(OP=='+')
        result=v1+v2;
    else if(OP=='-')
        result=v1-v2;
    else if(OP=='*')
        result=v1*v2;
    else if(OP=='/')
        result=v1/v2;
    else if(OP=='%')
        result=v1%v2;
}

```

```
        t1.setText(""+result);
    }
    if(str.equals("clear"))
    {
        t1.setText("");
    }
}
```

INPUT AND OUTPUT:





PRE LAB VIVA QUESTIONS:

1. what is awt and swing?
2. how do we create a jButton object?
3. what is a data type?
4. what is the purpose of break statement in switch case?

POST LAB VIVA QUESTIONS:

5. How to convert the string data into long data?
6. Which containers use a Border layout as their default layout?
7. What is an event and what are the models available for event handling?
8. Which listeners are implemented for JButton ?

WEEK-13: APPLETS

EXPERIMENT-13.1

OBJECTIVE:

To develop an Applet in java that displays a Simple Message

RESOURCES:

Net Beans, JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name Simple extends Applet.
3. Declare the Font class object.
4. Set the font.
5. Call the setFont method and drawString method `g.setFont(font);` `g.drawString("String", 50, 50);`

SOURCE CODE:

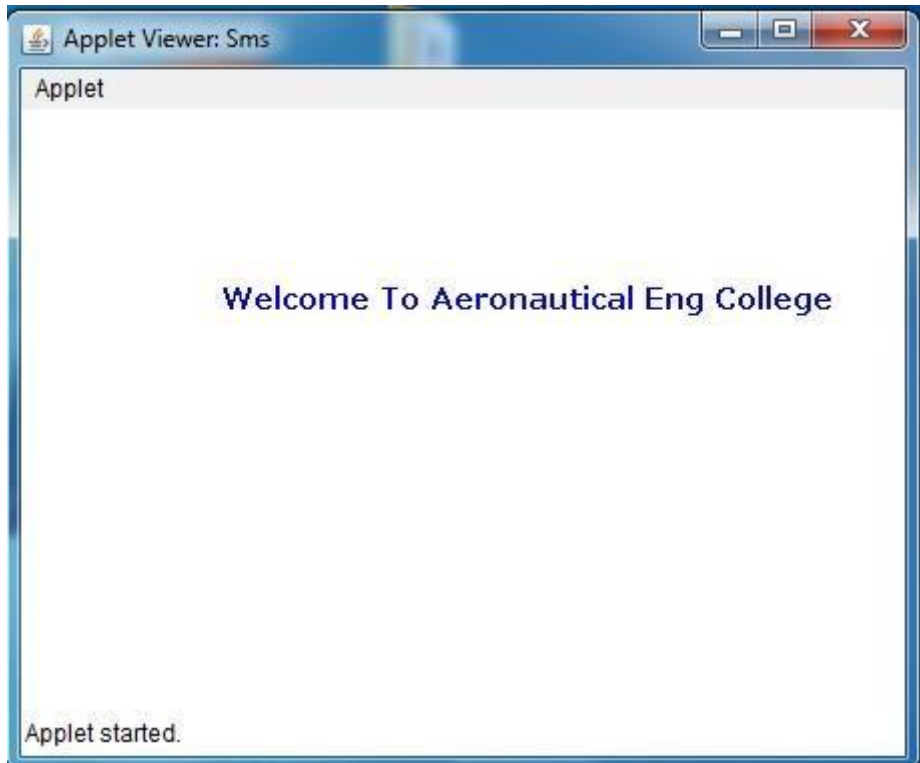
```
import java.applet.Applet;
import java.awt.*;

/*
   <applet code="Simple.class" width=300 height=300>
   </applet>
*/

public class Simple extends Applet
{
    public void init()
    {
    }

    public void paint(Graphics g)
    {
        g.setColor(Color.blue);
        Font font = new Font("verdana", Font.BOLD, 15);
        g.setFont(font);
        g.drawString("Welcome To Aeronautical Eng College", 50, 50);
    }
}
```

INPUT AND OUTPUT:



PRE LAB VIVA QUESTIONS:

1. Which containers use a Border layout as their default layout?
2. How are the elements of different layouts organized?
3. Which containers use a Flow layout as their default layout?
4. Can we create a sub package?

POST LAB VIVA QUESTIONS:

1. How does an applet will run?
2. Where does an applet will execute?
3. How do we create an applet?
4. What is the difference between application and applet?

EXPERIMENT-13.2

OBJECTIVE:

To develop an Applet in java that receives an integer in one Text Field, and computes its Factorial value and returns it in another text Field, when button named "Compute" is clicked.

RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name Fact extends Applet which implements ActionListener.
3. Declare the Label, TextField, Button variables.
4. Call init method
5. Call the setLayout method, setLayout(g);
6. Create the HTML file for applet tag.

SOURCE CODE:

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
/*
<applet code="Fact.class" width=300 height=300>
</applet>
*/
public class Fact extends Applet implements ActionListener
{
    Label l1, l2, l3;
    TextField tf1, tf2;
    Button b1;
    public void init()
    {
        setSize(400, 200);
        FlowLayout g = new FlowLayout();
        setLayout(g);
        l1 = new Label("Enter Value");
        l1.setForeground(Color.BLUE);
        add(l1);
        tf1 = new TextField(5);
        tf1.setText("0");
        add(tf1);
        b1 = new Button("Compute");
        b1.addActionListener(this);
        add(b1);
        l3 = new Label();
        add(l3);
        l2 = new Label("factorial: ");
```

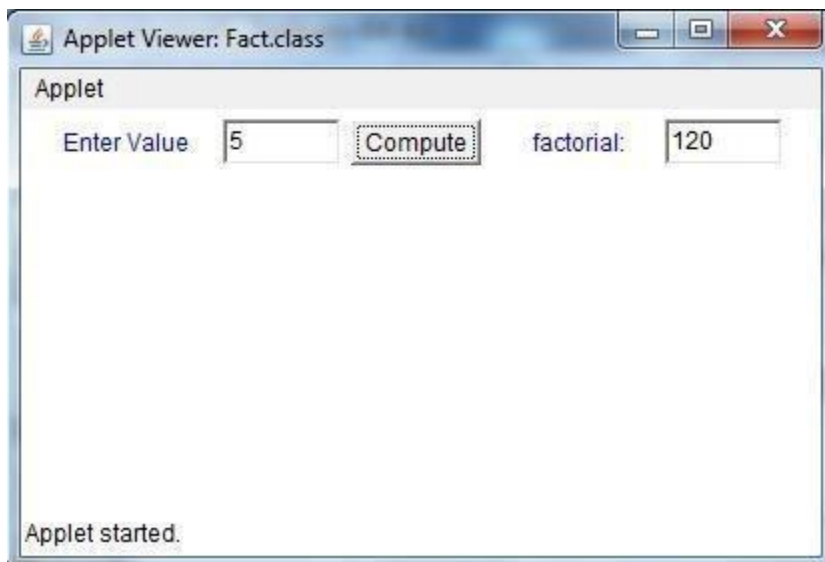


```

        l2.setForeground(Color.BLUE);
        add(l2);
        tf2 = new TextField(5);
        add(tf2);
    }
    public void actionPerformed(ActionEvent ae)
    {
        long n = Integer.parseInt(tf1.getText());
        long f = 1;
        while (n != 0)
        {
            f = f * n;
            n--;
        }
        tf2.setText(String.valueOf(f));
    }
}

```

10.2.6 INPUT AND OUTPUT:



PRE LAB VIVA QUESTIONS:

1. Can applets communicate with each other?
2. How does applet recognize the height and width?
3. How do you set security in applets?
4. What are different types of layouts used in java

POST LAB VIVA QUESTIONS:

5. Which events are extended for the button component?
6. What is an event and what are the models available for event handling?
7. What is meant by controls and what are different types of controls in AWT?
8. When do you use codebase in applet?

ADDITIONAL PROGRAMS

EXPERIMENT-1

OBJECTIVE:

Design a class to represent a Student details. Include the Student ID, name of the Student, branch, year and assign initial values, calculate average of marks of 6 subjects, calculate attendance percentage.

RESOURCES:

Net Beans, JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Create a class for student, which contains student ID, name, branch, year.
2. Define methods in that class to assign initial values, to calculate average of marks, to calculate percentage.

SOURCE CODE:

```
import java.util.Scanner;
class Student
{
    int id;
    String name;
    String branch;
    int year;
    void getData(int a,String b,String c,int d)
    {
        id=a;
        name=b;
        branch=c;
        year=d;
    }
    void Avg(int a,int b,int c,int d,int e,int f)
    {
        float avg;
        avg=(a+b+c+d+e+f)/6;
        System.out.println("Average is"+avg);
    }
    void Percentage(int a, int b)
    {
        float per;
        per=(a/b)*100;
        System.out.println("Percentage is"+per);
    }
}
```

```

class Execute
{
    public static void main(String args[])
    {
        Student st=new Student();
        Scanner s=new Scanner(System.in);
        System.out.println("enter ID,name,branch,year");
        st.getData(s.nextInt(),s.next(),s.next(),s.nextInt());
        System.out.println("enter 6 subject marks");
        st.Avg(s.nextInt(),s.nextInt(),s.nextInt(),s.nextInt(),s.nextInt(),s.nextInt());
        System.out.println("enter the present and total working days");
        st.Percentage(s.nextInt(), s.nextInt());
    }
}

```

OUTPUT:

```

Output
Bank (run) Bank (run) #2
run:
enter ID,name,branch,year
101 GGG CSE 2014
enter 6 subject marks
11 23 34 56 78 34
Average is39.0
enter the present and total working days
21 31
Percentage is0.0
BUILD SUCCESSFUL (total time: 1 minute 10 seconds)

```

PRE LAB VIVA QUESTIONS:

1. Define variable?
2. Define instance and static variable?
3. Define method and static method?
4. Define final variable?

POST LAB VIVA QUESTIONS:

5. Hat is an object and how to create object?
6. What is the difference between command line argument and a parameter?
7. What are different types of access specifier?
8. What is size of byte?

EXPERIMENT-2

OBJECTIVE:

Develop with suitable hierarchy, classes for Point, Shape, Rectangle, Square, Circle, Ellipse, Triangle, Polygon, etc. Design a simple test application to demonstrate dynamic polymorphism

RESOURCES:

Net Beans, JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM:

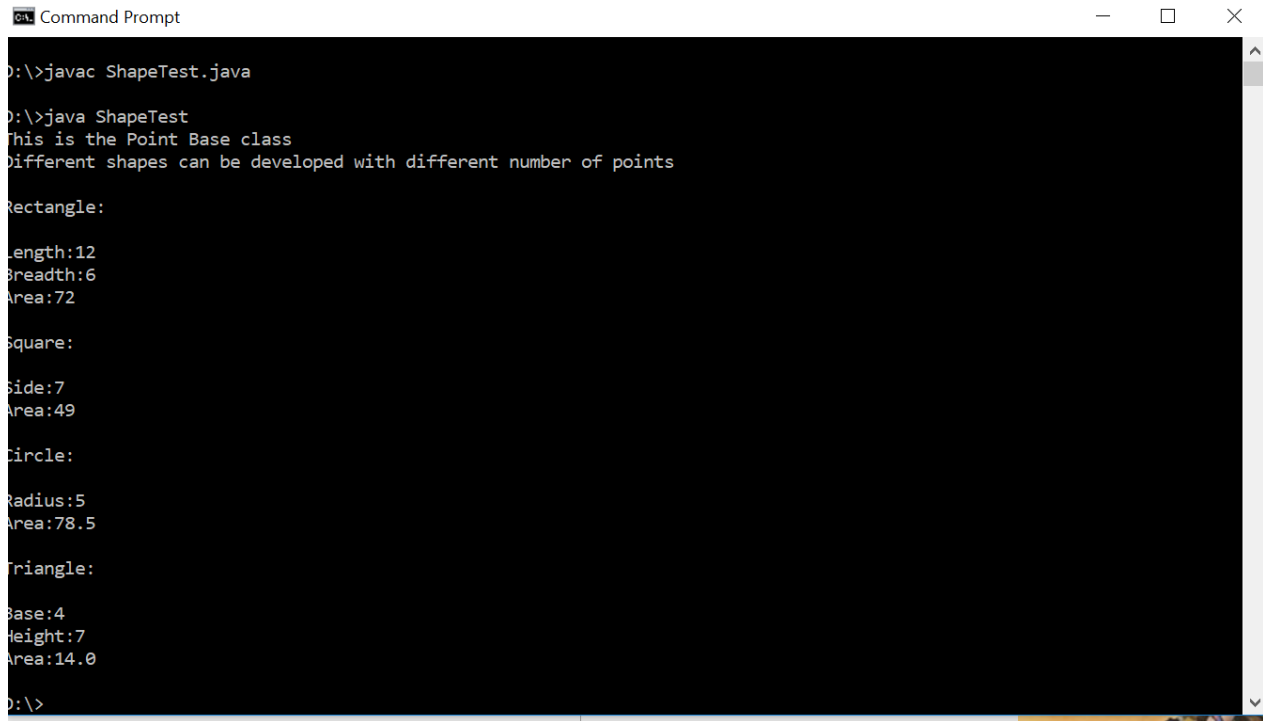
```
class point{
void show(){
System.out.println("This is the Point Base class");
}
}
class shape extends point{
void display(){
System.out.println("Different shapes can be developed
with different number of points");
}
}
class rectangle extends shape{
int l,b;
void getdata(int x,int y){
l=x;b=y;
}
void area(){
System.out.println("Length:"+l);
System.out.println("Breadth:"+b);
System.out.println("Area:"+(l*b));
}
}
class square extends shape{
int a;
void gdata(int x){
a=x;
}
void area(){
System.out.println("Side:"+a);
System.out.println("Area:"+(a*a));
}
}
class circle extends shape{
int r;
void get(int x){
r=x;
}
void area(){
```

```

System.out.println("Radius:"+r);
System.out.println("Area:"+(3.14*r*r));
}
}
class triangle extends shape{
int b,h;
void tdata(int x,int y){
b=x;h=y;
}
void area(){
System.out.println("Base:"+b);
System.out.println("Height:"+h);
System.out.println("Area:"+(0.5*b*h));
}
}
class ShapeTest{
public static void main(String args[]){
rectangle r = new rectangle();
square s = new square();
circle c = new circle();
triangle t = new triangle();
r.show();
s.display();
System.out.println("");
System.out.println("Rectangle:");
System.out.println();
r.getdata(12,6);
r.area();
System.out.println("");
System.out.println("Square:");
System.out.println();
s.gdata(7);
s.area();
System.out.println("");
System.out.println("Circle:");
System.out.println();
c.get(5);
c.area();
System.out.println("");
System.out.println("Triangle:");
System.out.println();
t.tdata(4,7);
t.area();
}
}

```

OUTPUT:



```
Command Prompt
D:\>javac ShapeTest.java
D:\>java ShapeTest
This is the Point Base class
Different shapes can be developed with different number of points
Rectangle:
Length:12
Breadth:6
Area:72
Square:
Side:7
Area:49
Circle:
Radius:5
Area:78.5
Triangle:
Base:4
Height:7
Area:14.0
D:\>
```

PRE LAB VIVA QUESTIONS:

1. Define class?
2. Define method overriding?
3. What is dynamic dispatch method?
4. Define reference variable?

POST LAB VIVA QUESTIONS:

5. What is abstract class?
6. What is run time polymorphism?
7. What is polymorphism?
8. What is compile time binding and run time binding?

EXPERIMENT-3

OBJECTIVE:

Write a program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.

RESOURCES:

Net Beans, JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Create three classes named by one, second, three.
2. All these three classes will implement Runnable interfaces
3. One class thread will print Goodmorning for every one second.sleep this thread for 1000 seconds.
4. Second class thread will print Hello for every two second.sleep this thread for 2000 seconds.
5. Third class thread will print World for every three second.sleep this thread for 3000 seconds.

SOURCE CODE:

```
public class JavaApplication1
{
    public static void main(String[] args)
    {
        One obj1=new One();
        Two obj2=new Two();
        Three obj3=new Three();
    }
}
```

```
class One implements Runnable
{
    One()
    {
        new Thread(this, "One").start();
        try
        {
            Thread.sleep(1000);
        }
        catch(InterruptedException e){}
    }
    public void run()
    {
        for ( int i=0;i<10 ;i++ )
        {
            try
```

```

    {
    Thread.sleep(1000);
    }
    catch(InterruptedException e){}
    System.out.println("Good Morning");
    }
}
class Two implements Runnable
{
    Two()
    {
        new Thread(this, "Two").start();
        try{
            Thread.sleep(2000);
        }
        catch(InterruptedException e){ }
    }
    public void run()
    {
        for (int i=0 ;i<10 ;i++ )
        {
            try
            {
                Thread.sleep(2000);
            }
            catch(InterruptedException e){ }
            System.out.println("Hello");
        }
    }
}
class Three implements Runnable
{
    Three()
    {
        new Thread(this, "Three").start();
        try{
            Thread.sleep(3000);
        }catch(InterruptedException e){ }
    }
    public void run()
    {
        for ( int i=0;i<10 ;i++ )
        {
            try{
                Thread.sleep(3000);
            }catch(InterruptedException e){ }
            System.out.println("Welcome");
        } }
}

```


OUTPUT:



```
Output
JavaApplication1 (run) JavaApplication1 (run) #2
Good Morning
Hello
Good Morning
Good Morning
Hello
Good Morning
Hello
```

PRE LAB VIVA QUESTIONS:

1. Define class and thread class?
2. Define runnable interface?
3. What is package of thread class?
4. Define interrupt?

POST LAB VIVA QUESTIONS:

5. What is multiprogramming?
6. Why class implement runnable interface?
7. How to current class thread name?
8. What is sleep() method?

EXPERIMENT-4

OBJECTIVE:

Write an Applet that computes the payment of a loan based on the amount of the loan, the interest rate and the number of months. It takes one parameter from the browser: Monthly rate; if true, the interest rate is per month, otherwise the interest rate is annual.

RESOURCES:

Net Beans, JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

PROGRAM LOGIC:

1. Start the program
2. Create a class with the name A implements ItemListener.
3. Create the FileInputStream object.
4. Create the Scanner object delimited by comma.
5. Close the JFrame.

SOURCE CODE:

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

/* <applet code = "LoanPayment" width=500 height=300 >
<param name = monthlyRate value=true>
</applet>
*/

public class LoanPayment extends Applet implements ActionListener
{
    TextField amt_t, rate_t, period_t;
    Button compute = new Button("Compute");
    boolean monthlyRate;

    public void init()
    {
        Label amt_l = new Label("Amount: ");
        Label rate_l = new Label("Rate: ", Label.CENTER); Label
        period_l = new Label("Period: ", Label.RIGHT);

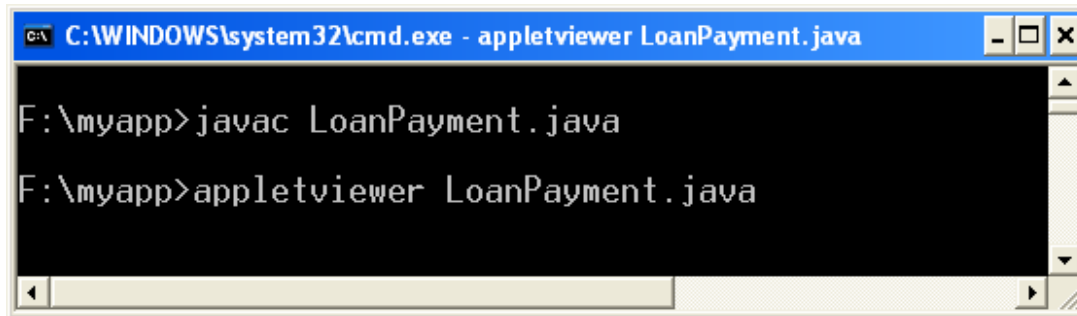
        amt_t = new TextField(10);
        rate_t = new TextField(10);
        period_t = new TextField(10);
```

```

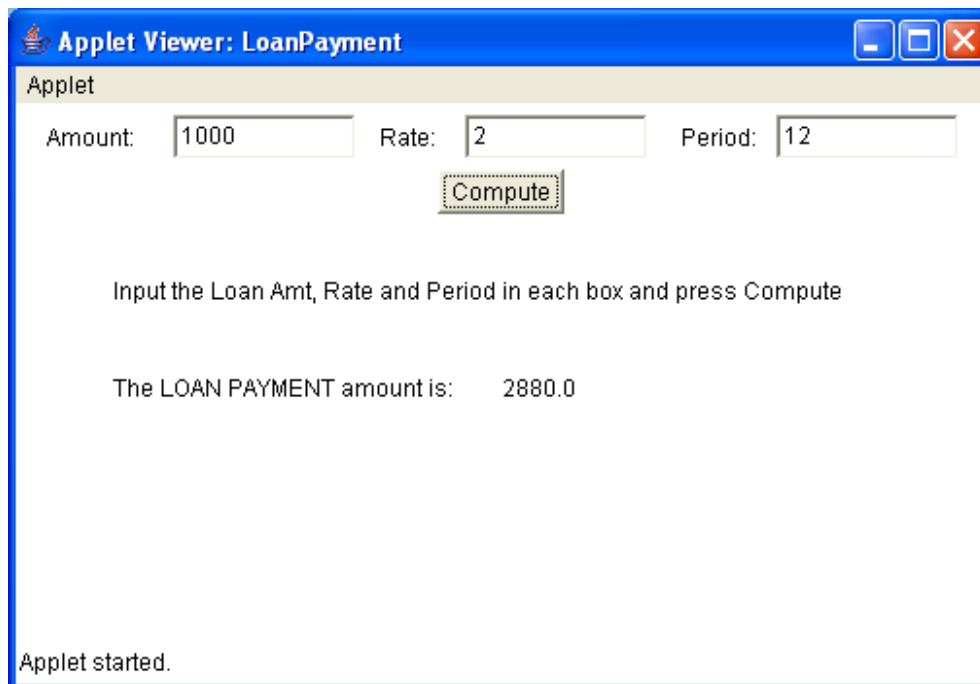
    add(amt_l);
    add(amt_t);
    add(rate_l);
    add(rate_t);
    add(period_l);
    add(period_t);
    add(compute);
    amt_t.setText("0");
    rate_t.setText("0");
    period_t.setText("0");
    monthlyRate = Boolean.valueOf(getParameter("monthlyRate"));
    amt_t.addActionListener(this);
    rate_t.addActionListener(this);
    period_t.addActionListener(this);
    compute.addActionListener(this);
}
public void paint(Graphics g)
{
    double amt=0, rate=0, period=0, payment=0;
    String amt_s, rate_s, period_s, payment_s;
    g.drawString("Input the Loan Amt, Rate and Period in each box and press Compute", 50,100);
    try
    {
        amt_s = amt_t.getText();
        amt = Double.parseDouble(amt_s);
        rate_s = rate_t.getText();
        rate = Double.parseDouble(rate_s);
        period_s = period_t.getText();
        period = Double.parseDouble(period_s);
    }
    catch (Exception e) { }
    if (monthlyRate)
        payment = amt * period * rate * 12 / 100;
    else
        payment = amt * period * rate / 100;
    payment_s = String.valueOf(payment);
    g.drawString("The LOAN PAYMENT amount is: ", 50, 150);
    g.drawString(payment_s, 250, 150);
}
public void actionPerformed(ActionEvent ae) {
    repaint();
}
}

```

4.4 INPUT & OUTPUT:



```
C:\WINDOWS\system32\cmd.exe - appletviewer LoanPayment.java
F:\myapp>javac LoanPayment.java
F:\myapp>appletviewer LoanPayment.java
```



PRE LAB VIVA QUESTIONS:

1. Define applet?
2. Define application program?
3. How to run applet program?
4. Define applet viewer?

POST LAB VIVA QUESTIONS:

1. What is applet tag?
2. Why html code is embedded in applet?
3. What is param tag?
4. What is sleep() method?

EXPERIMENT-5

OBJECTIVE:

Write a java programs to find factorial of a number. User is allowed to enter a number into the text field whose factorial is to be determined. On pressing the button the value of the text field is firstly converted into integer and then processed to find its factorial. The result will get displayed in another text field.(Hint: use swings).

RESOURCES:

Net Beans, JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

a. PROGRAM LOGIC:

1. Create a class FactorialApplet extends Applet implements ActionListener.
2. Create labels to enter a value and to print result.
3. Create two buttons to calculate and clear.

SOURCE CODE:

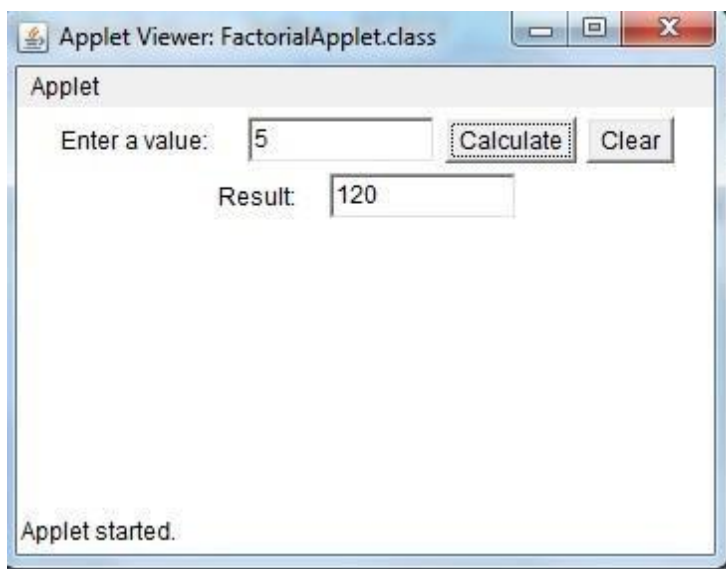
```
import java.applet.*;
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
/*<applet code="Factorial" width=500 height=500> </applet> */
public class FactorialApplet extends Applet implements ActionListener
{
    Label l1,l2;
    TextField t1,t2;
    Button b1,b2;
    public void init()
    {
        l1=new Label("Enter a value: ");
        l2=new Label("Result:");
        t1=new TextField(10);
        t2=new TextField(10);
        b1=new Button("Calculate");
        b2=new Button("Clear");
        add(l1);
        add(t1);
        add(b1);
        add(b2);
        add(l2);
        add(t2);
        b1.addActionListener(this);
        b2.addActionListener(this);
    }
}
```

```

    }
    public void actionPerformed(ActionEvent ae)
    {
        int n=Integer.parseInt(t1.getText());
        int fact=1;
        if(ae.getSource()==b1)
        {
            if(n==0||n==1)
            {
                fact=1;
                t2.setText(String.valueOf(fact));
            }
            else
            {
                for(int i=1;i<=n;i++)
                    fact=fact*i;
            }
            t2.setText(String.valueOf(fact));
        }
        else if(ae.getSource()==b2)
        {
            t1.setText("");
            t2.setText("");
        }
    }
}

```

OUTPUT:



PRE LAB VIVA QUESTIONS:

1. What is awt and swing?
2. How do we create a jButton object?
3. What is a data type?
4. What is the purpose of break statement in switch case?

POST LAB VIVA QUESTIONS:

5. How to convert the string data into long data?
6. Which containers use a BorderLayout as their default layout?
7. What is an event and what are the models available for event handling?
8. Which listeners are implemented for JButton?