



MARRI LAXMAN REDDY

INSTITUTE OF TECHNOLOGY & MANAGEMENT

(Affiliated to JNTU, Hyderabad, Approved by AICTE, New Delhi)



Accredited by NBA, Accredited by NAAC with 'A' Grade & Recognized Under section 2(f) & 12(B) of the UGC act, 1956

Dundigal (Vill .& Mandal), Medchal District, Hyderabad - 500043, Telangana.

JAVA PROGRAMMING LAB

MANUAL

Subject Code:CS408PC

Regulation:R18

II-B.TECH II SEMESTER

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that this manual is a bonafide record of practical work in the **JAVA PROGRAMMING** in II B.Tech II Sem (CSE) programme during the academic year 2019-20. This manual is prepared by **Mr. SIVA RAMA PRASAD KOLLU**(Asst.Professor).

PREFACE

This “**JAVA PROGRAMMING**” lab manual is intended to teach the basic Java Applications and Java Applets. Readers of this manual must be familiar with the basic syntax of C or C++ and Object Oriented features. Java is increasingly becoming the default choice of the IT industry especially industries involved in software development at system level. Therefore, for proper development of Java Applications among the students, this practical lab manual has been prepared. The manual contains the exercise programs and their solution for easy & quick understanding of the students. We hope that this practical manual will be helpful for students of CSE for understanding the subject from the point of view of applied aspects. There is always scope for improvement in the manual. We would appreciate to receive valuable suggestions from readers and users for future use.

By,
SIVA RAMA PRASAD KOLLU.

ACKNOWLEDGEMENT

It was really a good experience , working with “**JAVA PROGRAMMING**” . First we would like to thank Mr.K.Abdul Basith, Assoc.Professor, HOD of Department of Computer Science and Engineering, Marri Laxman Reddy Institute of technology & Management for his concern and giving the technical support in preparing the document. We are deeply indebted and gratefully acknowledge the constant support and valuable patronage of Dr.R.Kotaih, Director, Marri Laxman Reddy Institute of technology & Management for giving us this wonderful opportunity for preparing the “**JAVA PROGRAMMING**” laboratory manual. We express our hearty thanks to Dr.K.Venkateswara Reddy, Principal, Marri Laxman Reddy Institute of technology & Management, for timely corrections and scholarly guidance. At last, but not the least I would like to thanks the entire CSE and IT Department faculties those who had inspired and helped us to achieve our goal.

By,
SIVA RAMA PRASAD KOLLU.

INSTITUTION VISION AND MISSION

VISION

To be as an ideal academic institution by graduating talented engineers to be ethically strong, competent with quality research and technologies.

MISSION

To fulfill the promised vision through the following strategic characteristics and aspirations:

- Utilize rigorous educational experiences to produce talented engineers.
- Create an atmosphere that facilitates the success of students.
- Programs that integrate global awareness, communication skills and Leadership qualities.
- Education and Research partnership with institutions and industries to prepare the students for interdisciplinary research.

DEPARTMENT VISION AND MISSION**VISION**

To empower the students to be technologically adept, innovative, self-motivated and responsible global citizen possessing human values and contribute significantly towards high quality technical education with ever changing world.

MISSION

- To offer high-quality education in the computing fields by providing an environment where the knowledge is gained and applied to participate in research, for both students and faculty.
- To develop the problem solving skills in the students to be ready to deal with cutting edge technologies of the industry.
- To make the students and faculty excel in their professional fields by inculcating the communication skills, leadership skills, team building skills with the organization of various cocurricular and extra-curricular programmes.
- To provide the students with theoretical and applied knowledge, and adopt an education approach that promotes lifelong learning and ethical growth.

PROGRAMME EDUCATIONAL OBJECTIVES

PROGRAMME EDUCATIONAL OBJECTIVES

The Programme Educational Objectives (PEOs) that are formulated for the CSE programme are listed below:

PEO1: Establish a successful professional career in industry, government or academia.

PEO2: Gain multidisciplinary knowledge providing a sustainable competitive edge in higher studies or Research.

PEO3: Promote design, analyze, and exhibit of products, through strong communication, leadership and ethical skills, to succeed an entrepreneurial.

PROGRAM SPECIFIC OUUTCOMES

PSO1: Applications of Computing: Ability to use knowledge in various domains to provide solution to new ideas and innovations.

PSO2: Programming Skills: Identify required data structures, design suitable algorithms, develop and maintain software for real world problems.

PROGRAMME OUT COMES

The Program Outcomes (POs) of the department are defined in a way that the Graduate Attributes are included, which can be seen in the Program Outcomes (POs) defined.

Program Outcomes (POs) department are as stated below:

- a : An ability to apply knowledge of Science, Mathematics, Engineering & Computing fundamentals for the solutions of Complex Engineering problems.
- b : An ability to identify, formulates, research literature and analyze complex engineering problems using first principles of mathematics and engineering sciences.
- c : An ability to design solutions to complex process or program to meet desired needs.
- d : Ability to use research-based knowledge and research methods including design of experiments to provide valid conclusions.
- e : An ability to use appropriate techniques, skills and tools necessary for computing practice.
- f : Ability to apply reasoning informed by the contextual knowledge to assess social issues, consequences & responsibilities relevant to the professional engineering practice.
- g : Ability to understand the impact of engineering solutions in a global, economic, environmental, and societal context with sustainability.
- h : An understanding of professional, ethical, Social issues and responsibilities.
- i : An ability to function as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.
- j : An ability to communicate effectively on complex engineering activities within the engineering community.
- k : Ability to demonstrate and understanding of the engineering and management principles as a member and leader in a team.
- l : Ability to engage in independent and lifelong learning in the context of technological change.

COURSE STRUCTURE, OBJECTIVES & OUTCOMES

COURSE STRUCTURE:

Laboratory subjects – Internal and external evaluation – Details of marks **JAVA PROGRAMMING** lab will have a continuous evaluation during 3rd semester for 25 sessional marks and 75 end semester examination marks. Out of the 25 marks for internal evaluation, day-to-day work in the laboratory shall be evaluated for 15 marks and internal practical examination shall be evaluated for 10 marks conducted by the laboratory teacher concerned. The end examination will be evaluated for a maximum of 75 marks. The end semester examination shall be conducted with an external examiner and internal examiner. The external examiner shall be appointed by the principal / Chief Controller of examinations

COURSE OBJECTIVES:

- To write programs using abstract classes.
- To write programs for solving real world problems using java collection frame work.
- To write multithreaded programs.
- To write GUI programs using swing controls in Java.
- To introduce java compiler and eclipse platform.
- To impart hands on experience with java programming.

COURSE OUTCOMES:

- Able to write programs for solving real world problems using java collection frame work.
- Able to write programs using abstract classes.
- Able to write multithreaded programs.
- Able to write GUI programs using swing controls in Java.

LIST OF EXPERIMENTS

1. Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a 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 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 exceptions like divided by zero
3. Write a Java Program to
 - a. Develop an applet in Java that displays a simple message.
 - b. 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 the button named “Compute” is clicked.
4. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, 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.
5. 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.
6. Write a Java program for the following:
 - i. Create a doubly linked list of elements.
 - ii. Delete a given element from the above list.
 - iii. Display the contents of the list after deletion.
7. 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.
8. 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.
9. 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 to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.
10. 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).

11. 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).
12. Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.
13. Write a Java program to list all the files in a directory including the files present in all its subdirectories.
14. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order
15. Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.

Week-1

Aim: Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

Aim: Write a java program to find the factorial of the number.

Program:

```
class Fact
{
    public static void main(String[] args)
    {
        int number = 5;
        int factorial = number;
        for(int i =(number - 1); i > 1; i--)
        {
            factorial = factorial * i;
        }
        System.out.println("Factorial of a number is:" + factorial);
    }
}
```

Output: Factorial of a number is:120

Aim :Write a java program to display Fibonacci series .

Program:

```
import java.util.Scanner;
public class Fibonacci
{
    public static void main(String[] args)
    {
        int n, a = 0, b = 0, c = 1;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter value of n:");
        n = s.nextInt();
        System.out.print("Fibonacci Series:");
        for(int i = 1; i <= n; i++)
        {
            a = b;
            b = c;
            c = a + b;
            System.out.print(a+" ");
        }
    }
}
```

Output:

```
Enter value of n:5
Fibonacci Series:01123
```

Aim : Write a java program to find whether the give number is Armstrong number or not.

Program:

```
import java.io.*;
import java.util.*;
class ArmStrong
{
    public static void main(String[] args)throws IOException
    {
        int t,s=0,n,r;
        System.out.println("Enter number:");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        t=n;
        while( t!=0)
        {
            r=t% 10;
            s=s+r*r*r;
            t=t/10;
        }
        if(n==s)
            System.out.println("Number given is Armstrong");
        else
            System.out.println("Number given is not Armstrong");
    }
}
```

Output:

```
Enter number:153
Number given is Armstrong
```

Aim :Write a java program to reverse the given string.

Program:

```
import java.io.*;
import java.util.*;
class RevStr
{
    public static void main(String[] args) throws IOException
    {
        String or, rev="";
        DataInputStream dis=new DataInputStream(System.in);
        System.out.println("Enter String:");
        or=dis.readLine();
        int len=or.length();
        for(int i=len-1;i>=0;i--)
        {
            rev=rev+or.charAt(i);
        }
        System.out.println("Reverse of a string:"+rev);
    }
}
```

Output:

```
Enter String:siva
Reverse of a string:avis
```

Aim: Write a java program to evaluate quadratic equation

Program:

```
import java.io.*;
class QudEq
{
    public static void main(String args[])throws IOException
    {
        BufferedReader br;
        int a,b,c;
        double x,y,z;
        System.out.println("ENTER a VALUE:");
        br= new BufferedReader(new InputStreamReader(System.in));
        a=Integer.parseInt(br.readLine());
        System.out.println("ENTER b VALUE:");
        b=Integer.parseInt(br.readLine());
        System.out.println("ENTER c VALUE:");
        c=Integer.parseInt(br.readLine());
        z=Math.sqrt(b*b-4*a*c);
        x=(-b+Math.sqrt(b*b-4*a*c))/(2*a);
        y=(-b-Math.sqrt(b*b-4*a*c))/(2*a);
        if(z>0)
        {
            System.out.println("THE REAL SOLUTIONS ARE =" +x+"\t"+"Y="+y);
        }
        else
        {
            System.out.println("THERE ARE NO REAL SOLUTIONS");
        }
    }
}
```

Output:

Enter a value:

4

Enter b value:

6

Enter c value:

-2

THE REAL SOLUTIONS ARE X=0.28077640640441515 Y=-1.7807764064044151

Aim :Write a java program to print Fibonacci series for the given number using recursion

Program:

```
import java.io.*;
public class FibRec
{
    static int fib(int i)
    {
        if(i==1||i==2)
        {
            return i-1;
        }
        else
        {
            return fib(i-1)+fib(i-2);
        }
    }
}

public static void main(String args[])throws IOException
{
    BufferedReader br;
    int n;
    System.out.println("enter n value:");
    br = new BufferedReader(new InputStreamReader(System.in));
    n=Integer.parseInt(br.readLine());
    System.out.print("FIBONACCI SEQUENCE:");
    for(int i=2;i<=n+1;i++)
    {
        System.out.print(fib(i)+"\t");
    }
}
}
```

Output:

enter n value:

5

FIBNOCCI SEQUENCE:1 1 2 3 5

Aim: Write a java program to print Fibonacci series for the given number using non-recursion

Program:

```
import java.io.*;
class FibNonRec
{
    public static void main(String args[])throws IOException
    {
        BufferedReader br;
        int a=0;
        int b=1,t,n;
        br=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("enter n value:");
        n=Integer.parseInt(br.readLine());
        System.out.print("FIBONACCI SEQUENCE IS:");
        System.out.print(a+"\t");
        do
        {
            System.out.print(b+"\t");
            t=b;
            b=a+b;
            a=t;
        }while(b<=n);
    }
}
```

Output:

enter n value:

5

FIBNOCCI SEQUENCE IS: 0 1 1 2 3 5

Aim: Write a java program to count number of words in a given text

Program:

```
import java.io.*;
import java.util.*;
class CountWords
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter a string : ");
        String str=br.readLine();
        int count=0;
        if(str.length()==0)
        {
            System.out.println("No.of words in given text:" + count);
        }
        else
        {
            for(int i=0;i<str.length();i++)
            {
                if(str.charAt(i)==' ')
                {
                    count++;
                }
            }
            System.out.println("No.of words in given text:" + (count+1));
        }
    }
}
```

Output:

Enter a string:

hi this is siva

No.of words in given text:

4

Aim: Write a java program to check the given string is palindrome or not

Program:

```
import java.io.*;
public class Palindrome
{
    public static void main(String args[]) throws IOException
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter String:");
        String str=br.readLine();
        char ch[]=new char[str.length()];
        for(int i=str.length()-1,j=0;i>=0;i--,j++)
            ch[j]=str.charAt(i);
        String restr=new String(ch);
        System.out.println("Reverse of String "+str+" is "+restr);
        if(str.compareTo(restr)==0)
            System.out.println(str+" "+"is a Palindrome");
        else
            System.out.println(str+" "+"is not a Palindrome");
    }
}
```

Output:

Enter String:

madam

Reverse of String madam is madam

madam is a Palindrome

Aim: Write a java program to print the prime numbers upto nth number

Program:

```
import java.io.*;
class Primenos
{
    public static void main(String args[])throws IOException
    {
        BufferedReader br;
        int i,t,flag,n;
        br=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("ENTER n VALUE:");
        n=Integer.parseInt(br.readLine());
        System.out.println("PRIME NUMBERS UP TO"+" "+n+":");
        for(i=2;i<=n;i++)
        {
            flag=1;
            for(t=2;t<i;t++)
            {
                if(i%t==0)
                {
                    flag=0;
                    break;
                }
            }
            if(flag==1)
                System.out.println(i);
        }
    }
}
```

Output:

ENTER n VALUE:

5

PRIME NUMBERS UP TO 5:

2

3

5

Aim: To read a line of integers and display each integer and sum of all integers

Program:

```
import java.util.*;
import java.io.*;
class StrTok
{
    public static void main(String args[])throws IOException
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter String in numbers");
        String str=br.readLine();
        StringTokenizer s=new StringTokenizer(str);
        int x,sum=0;
        while(s.hasMoreTokens())
        {
            String str1=s.nextToken();
            x=Integer.parseInt(str1);
            System.out.println(x);
            sum+=x;
        }
        System.out.println("total sum:"+sum);
    }
}
```

Output:

```
Enter String in numbers
3 8 5
3
8
5
total sum: 16
```

Viva Questions:

1. What do you know about Java?
2. What are the supported platforms by Java Programming Language?
3. List any five features of Java?
4. Why is Java Architectural Neutral?
5. How Java enabled High Performance?
6. Why Java is considered dynamic?
7. What is Java Virtual Machine and how it is considered in context of Java's platform independent feature?
8. List any two Java IDE's?
9. List some of the Java keywords.
10. What do you mean by Object?
11. Define class?
12. What kind of variables a class can consist of?
13. What is a Local Variable?
14. What is a Instance Variable?
15. What is a Class Variable?
16. What is the default value of byte datatype in Java?
17. What is the default value of float and double datatype in Java?
18. When a byte datatype is used?
19. What is a static variable?
20. When parseInt() method can be used?
21. Define Inheritance?
22. What is Polymorphism?
23. What is Abstraction?
24. What is Encapsulation?
25. What is the primary benefit of Encapsulation?
26. Explain the following line used under Java Program

```
public static void main (String args[ ])
```
27. Define JRE i.e. Java Runtime Environment?
28. Define JIT compiler?
29. What is the difference between object oriented programming language and object based programming language?
30. How many bits are used to represent Unicode, ASCII, UTF-16, and UTF-8 characters?
31. What is the difference between static and non-static variables?
32. What is the difference between a break statement and a continue statement?
33. What is the purpose of the System class?
34. List primitive Java types?
35. Which arithmetic operations can result in the throwing of an ArithmeticException?
36. Variable of the boolean type is automatically initialized as?
37. What will happen if static modifier is removed from the signature of the main method?
38. What is the default value of an object reference declared as an instance variable?
39. What is dot operator?

40. What is type casting?
41. What is the difference between the >> and >>> operators?
42. What is the range of the short type?
43. Which number is denoted by leading zero in java?
44. Which number is denoted by leading 0x or 0X in java?
45. Break statement can be used as labels in Java?
46. Is there any need to import java.lang package?
47. What is difference between Path and Classpath?
48. Can a class declared as private be accessed outside it's package?
49. List out primitive data types in java.
50. What is the size of short data type?

Week 2:

Aim: 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 exceptions like divided by zero.

Program:

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

/*
<applet code=" Calculator" width=500 height=500>
</applet>
*/

public class Calculator extends Applet implements ActionListener
{
    double v1,v2,result;
    TextField t1;
    Button b[]=new Button[10];
    Button add,sub,mul,div,clear,mod,EQ;
    char OP;
    public void init()
    {
        setBackground(Color.white);
        t1=new TextField(10);
        GridLayout gl=new GridLayout(6,3);
        setLayout(gl);
        for(int i=0;i<10;i++)
        {
            b[i]=new Button(""+i);
        }
        add=new Button("+");
        sub=new Button("-");
        mul=new Button("*");
        div=new Button("/");
        mod=new Button("%");
        clear=new Button("clear");
        EQ=new Button("=");
        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("+"))
    {
        v1=Integer.parseInt(t1.getText());
        OP='+';
        t1.setText("");
    }
    else if(str.equals("-"))
    {
        v1=Integer.parseInt(t1.getText());
        OP='-';
        t1.setText("");
    }
    else if(str.equals("*"))
    {
        v1=Integer.parseInt(t1.getText());
        OP='*';
        t1.setText("");
    }
    else if(str.equals("/"))

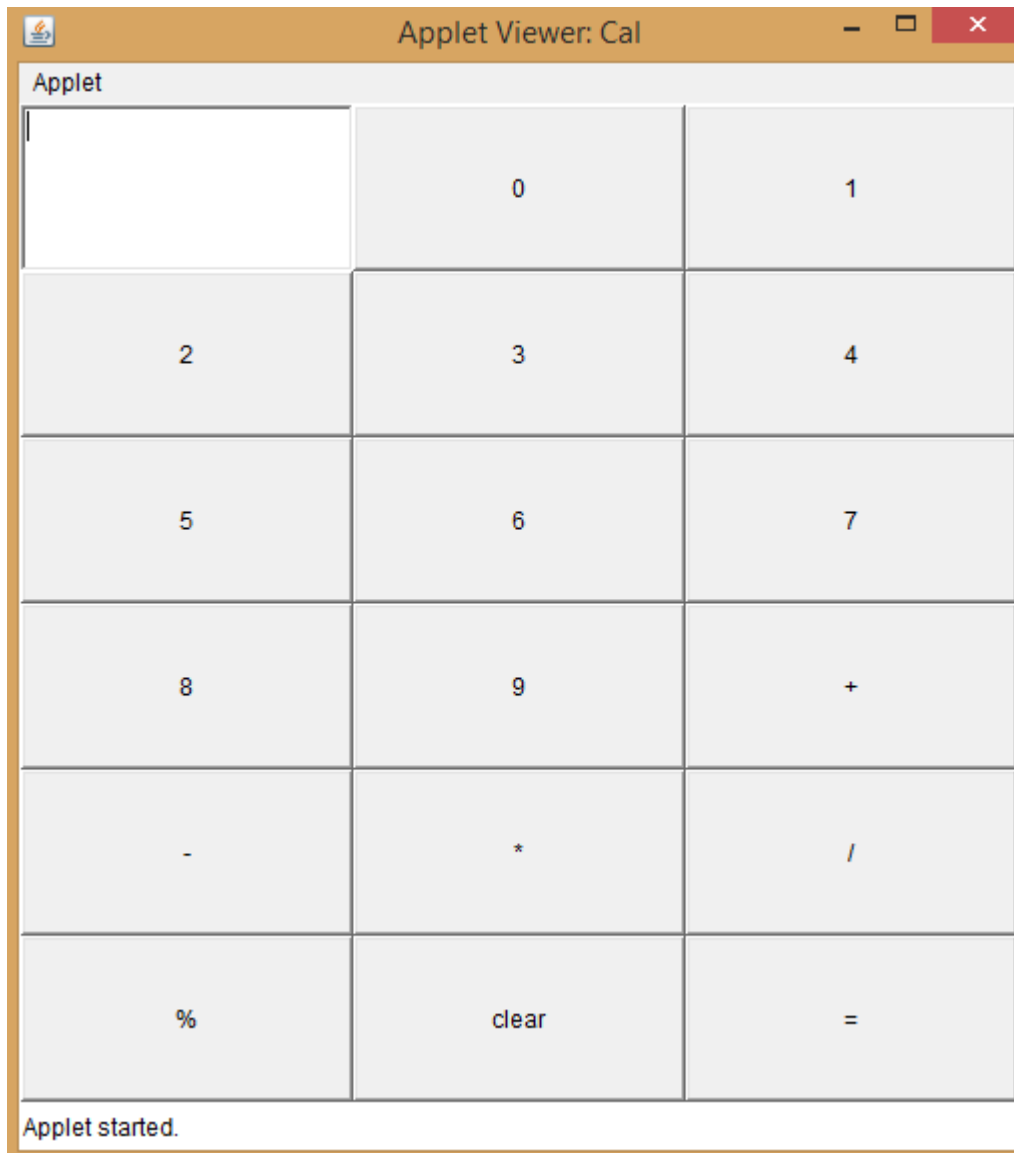
```

```

        {
            v1=Integer.parseInt(t1.getText());
            OP='/';
            t1.setText("");
        }
    else if(str.equals("%"))
    {
        v1=Integer.parseInt(t1.getText());
        OP='%';
        t1.setText("");
    }
    if(str.equals("="))
    {
        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("");
    }
}
}
}

```

Output:



Viva Questions:

1. What is the difference between an Applet and a Java Application ?
2. How to create a TextField?
3. How to create a Label?
4. How to create a Button?
5. To which package Applet class belongs to?
6. What is the use of ActionListener?
7. Write the syntax of applet tag.
8. List out some of the classes of awt package.
9. List out the methods of Applet class.
10. what is the use of init() method in an applet?
11. How to set background of an applet?
12. List out some colors of Color class.
13. What is meant by GridLayout?
14. How to set layout for an applet?
15. Where does actionPerformed() method is present?
16. What is meant by this keyword?
17. Write the syntax for creation of Button.
18. How to set name for a button?
19. How to set name for a label?
20. How to set length of the text field?
21. What is method used to add the listener for a component?
22. How to add components to the applet?
23. What is method used to remove the listener for a component?
24. Write the syntax of actionPerformed() method.
25. What is the use of getActionCommand() method?
26. What is the return type of charAt() method?
27. Write the syntax of charAt() method.
28. In which package Character class is present?
29. What is the use of isDigit() method?
30. What is the syntax of getText() method?
31. What is the syntax of setText() method?
32. How to use equals() method?
33. What is the return type of equals() method?
34. To which class equals() method belongs to?
35. What is the use of parseInt() method?
36. What is the super class of all the applets?
37. How to set number of rows and columns in GridLayout?
38. Where does event subpackage present?
39. what do you mean by </> in applet tag?
40. What are the methods of ActionListener?
41. What is access specifier for actionPerformed() method?
42. Why public is access specifier for actionPerformed() method?

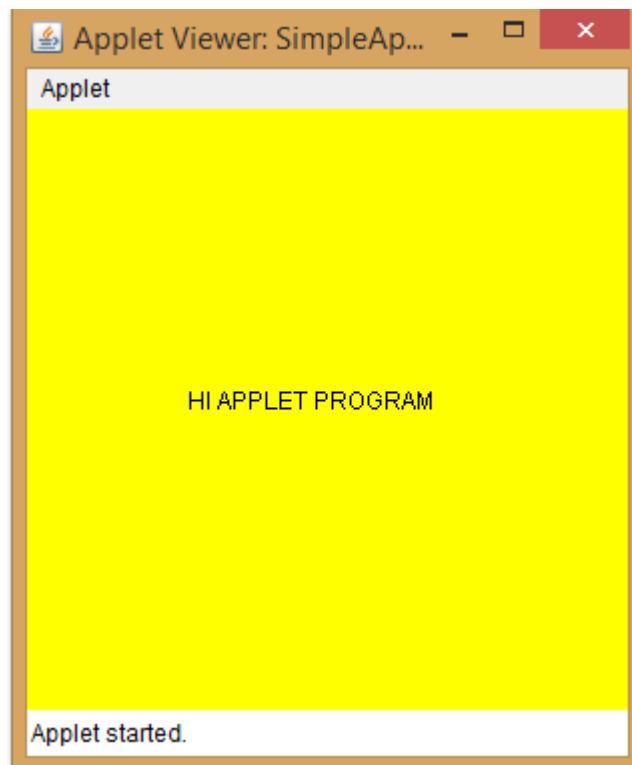
43. What is the use of this keyword?
44. Write any one constructor of Button class.
45. What is the use of ActionEvent?
46. How to declare a group of buttons which are in sequence?
47. Which component is required to print the output of the above program?
48. What is the common modifier for all the colors in the Color class?
49. What is the output of above program if we divide a number with zero?
50. When the output of the above program will be NaN?

Week 3

a) **Aim:** To Write a java program for simple applet program

Program:

```
import java.applet.*;
import java.awt.*;
/*
<applet code="SimpleApplet" height=300 width=300>
</applet>
*/
public class SimpleApplet extends Applet
{
    public void paint(Graphics g)
    {
        g.setColor(Color.black);
        setBackground(Color.yellow);
        g.drawString("HI APPLET PROGRAM",80,150);
    }
}
```

Output:

b) Aim: To develop an applet in java that receives an integer in one text field and computes its factorial in value and returns it in another text field ,when the button named “Compute” is clicked.

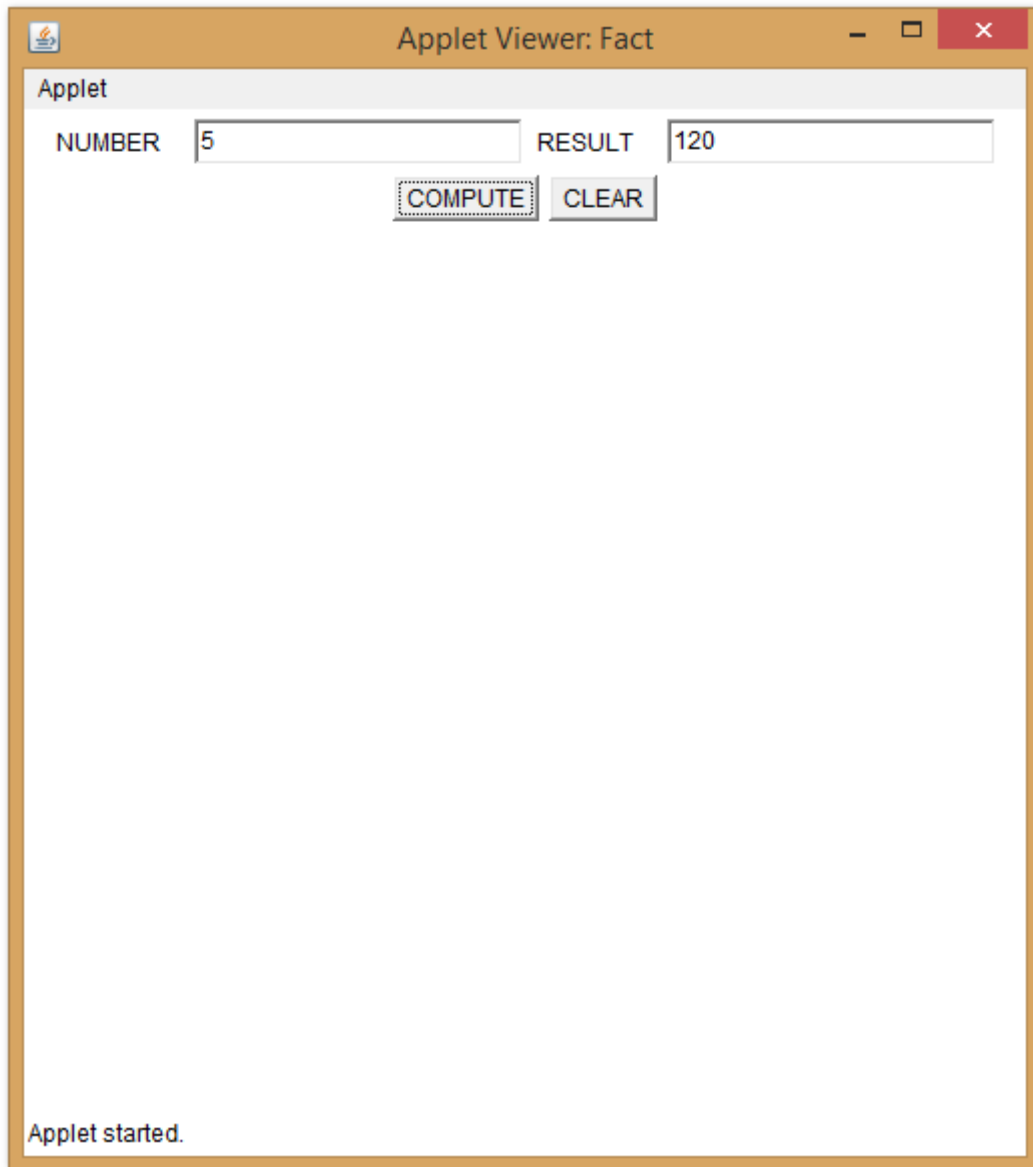
Program:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code=Fact width=500 height=500>
</applet>*/
public class Fact extends Applet implements ActionListener
{
    Button b1,b2;
    Label l1,l2;
    TextField tf1,tf2;
    public void init()
    {
        b1=new Button("COMPUTE");
        b1.addActionListener(this);
        b2=new Button("CLEAR");
        b2.addActionListener(this);
        tf1=new TextField(20);
        tf2=new TextField(20);
        l1=new Label("NUMBER");
        l2=new Label("RESULT");
        // setLayout(new GridLayout(3,2));
        add(l1);
        add(tf1);
        add(l2);
        add(tf2);
        add(b1);
        add(b2);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==b1)
        {
            int a=Integer.parseInt(tf1.getText());
            int fact=1;
            for(int i=1;i<=a;i++)
                fact*=i;
            tf2.setText(""+fact);
        }
        else
        {
            tf1.setText("");
        }
    }
}
```



```
        tf2.setText("");  
    }  
}
```

Output:



Viva Questions:

1. What is the use of applet package?
2. What is the use of awt package?
3. What is the use of height field in applet code?
4. What is the use of width field in applet code?
5. How to set height of an applet?
6. How to compile an applet?
7. How to execute an applet?
8. Why awt package is used?
9. Why java interpreter is not used to execute an applet?
10. How to execute an applet using html file?
11. What is the default layout of an applet?
12. How to implement the java code on applet?
13. How many class can have public in a program?
14. In how many ways we can execute an applet?
15. Write the procedure to execute an applet in vi editor.
16. What are the steps involved in executing an applet by using .html file?
17. where the java code will be included in the applet code?
18. Why main() method is not used in applet program?
19. Will applet interfere with the file system of a computer?
20. Is it secure to use applets?
21. What is the syntax of paint() method?
22. To which class paint() method belongs to?
23. Why the paint() method is public?
24. What is the use of Graphics class?
25. To which package Graphics class belongs to?
26. What is the parameter of the paint() method?
27. How to create Graphics reference in the paint() method?
28. What is the use of drawstring() method?
29. What are the parameters of drawstring() method?
30. How to call drawstring() method?
31. What is the parameter of setColor() method?
32. What is the use of start() method of an applet?
33. How to terminate applet?
34. What is the method used to terminate an applet?
35. What are the attributes of applet tag?
36. How to determine the width and height of an applet?
37. Can You Write A Java Class That Could Be Used Both As An Applet As Well As An Application?

38. How many ways exist to display messages to the user in applets?
39. What is the tag used to embed an applet in HTML file?
40. What is the importance of applets in Java coding?
41. How will you initialize an applet?
42. What is the order of method invocation in an applet?
43. What is the sequence for calling the methods by AWT for applets?
44. What tags are mandatory when creating HTML to display an applet?
45. Which method is used to output a string to an applet?
46. Is drawstring() method overloaded?
47. To which method drawstring () method belongs to?
48. To which class setBackground() method belongs to?
49. What is an Applet ?
50. What happens when an applet is loaded ?

Week-4

Aim: Write a Java program that creates a user interface to perform integer divisions. 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 or Num2 were not an integer, 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.

Program:

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

class SampleDialog extends Dialog implements ActionListener
{
    SampleDialog(Frame parent,String title,String msg)
    {
        super(parent,title,false);
        setLayout(new FlowLayout());
        setSize(300,200);
        add(new Label(msg));
        Button b;
        add(b=new Button("OK"));
        b.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae)
    {
        dispose();
    }
}

public class Division extends Frame implements ActionListener
{
    Button b1,b2;
    Label l1,l2,l3;
    TextField tf1,tf2,tf3;
    Division()
    {
        super("Exception Handler");
        b1=new Button("DIVIDE");
        b1.addActionListener(this);
        b2=new Button("CLEAR");
        b2.addActionListener(this);
        tf1=new TextField(15);
        tf2=new TextField(15);
        tf3=new TextField(20);
        l1=new Label("Enter the Numerator");
```

```

l2=new Label("Enter the Denaminator");
l3=new Label("RESULT");
setLayout(new FlowLayout());
add(l1);
add(tf1);
add(l2);
add(tf2);
add(l3);
add(tf3);
add(b1);
add(b2);
addWindowListener (new WindowAdapter() {
    public void windowClosing (WindowEvent e)
    {
        System.exit(0);
    }
});
}

public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==b1)
    {
        try
        {
            int a=Integer.parseInt(tf1.getText());
            int b=Integer.parseInt(tf2.getText());
            int c=a/b;
            tf3.setText(""+c);
        }

        catch(ArithmeticException ex)
        {
            tf3.setText("--");
            SampleDialog d=new SampleDialog(this,"EXCEPTION","DIVIDE BY ZERO");
            d.setVisible(true);
        }
        catch(Exception ex)
        {
            SampleDialog d=new SampleDialog(this,"EXCEPTION",""+ex);
            d.setVisible(true);
        }
    }
    else
    {

```

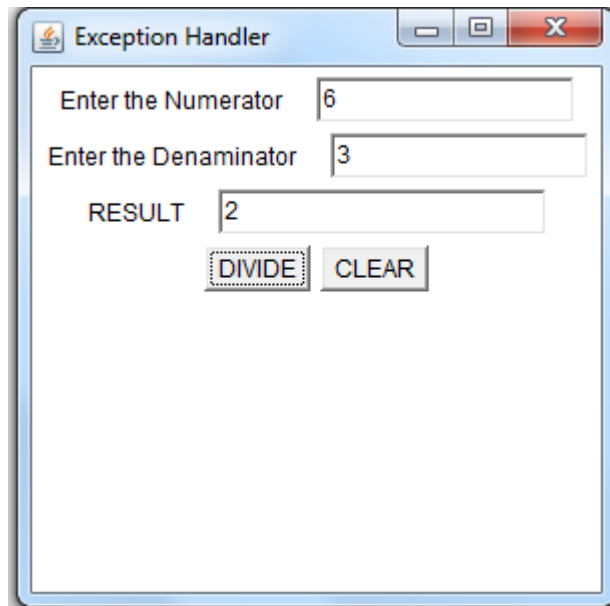
```

        tf1.setText("");
        tf2.setText("");
        tf3.setText("");
    }
}

public static void main(String args[])
{
    Division b=new Division();
    b.setSize(300,300);
    b.setVisible(true);
}
}

```

Output:



Viva Questions:

1. what are static blocks and static initializers in Java ?
2. How to call one constructor from the other constructor ?
3. What is method overriding in java ?
4. What is super keyword in java ?
5. Difference between method overloading and method overriding in java.
6. Difference between abstract class and interface ?
7. Why java is platform independent?
8. What is method overloading in java ?
9. What is difference between c++ and Java ?
10. What is JIT compiler ?
11. What is bytecode in java ?
12. Difference between this() and super() in java ?
13. What is a class ?
14. What is an object ?
15. What is method in java ?
16. What is encapsulation ?
17. Why main() method is public, static and void in java ?
18. Explain about main() method in java ?
19. What is constructor in java ? .
20. What is difference between length and length() method in java ?
21. What is ASCII Code?
22. What is Unicode ? .
23. Difference between Character Constant and String Constant in java ? .
24. What are constants and how to create constants in java? .
25. Difference between '>>' and '>>>' operators in java?
26. Explain Java Coding Standards for classes or Java coding conventions for classes?
27. Explain Java Coding standards for interfaces?
28. Explain Java Coding standards for Methods?
29. Explain Java Coding Standards for variables ?
30. Explain Java Coding Standards for Constants?
31. Difference between overriding and overloading in java?
32. What is 'IS-A' relationship in java?
33. What is 'HAS A' relationship in java?
34. Difference between 'IS-A' and 'HAS-A' relationship in java.
35. Explain about instanceof operator in java?
36. What does null mean in java?
37. What are identifiers in java?
38. What are access modifiers in java?
39. What is the difference between access specifiers and access modifiers in java?
40. What access modifiers can be used for class ?
41. Explain what access modifiers can be used for method?
42. Explain what access modifiers can be used for variables?
43. What is final access modifier in java?

44. Explain about abstract classes in java.
45. Can we create constructor in abstract class ?
46. What is an exception in java?
47. State some situations where exceptions may arise in java?
48. What is Exception handling in java?
49. What is an error in Java?
50. What are advantages of Exception handling in java?

Week-5

Aim: 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.

Program:

```
import java.util.*;
class Even implements Runnable
{
    public int x;
    public Even(int x)
    {
        this.x = x;
    }
    public void run()
    {
        System.out.println("New Thread "+ x +" is EVEN 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("New Thread "+ x +" is ODD and Cube of " + x + " is: " + x *
x * x);
    }
}
class A extends Thread
{
    public void run()
    {
        int num = 0;
        Random r = new Random();
        try
        {
            for (int i = 0; i < 5; i++)
            {
```


Viva Questions:

- 1) What is Thread in java?
- 2) What is difference between Process and Thread in java?
- 3) How to implement Threads in java?
- 4) What is ThreadGroup in java, What is default priority of newly created threadGroup, mention some important ThreadGroup methods ?
- 5) We should implement Runnable interface or extend Thread class. What are differences between implementing Runnable and extending Thread?
- 6) How can you say Thread behaviour is unpredictable?
- 7) When threads are not lightweight process in java?
- 8) How can you ensure all threads that started from main must end in order in which they started and also main should end in last?
- 9) What is difference between starting thread with run() and start() method?
- 10) What is significance of using Volatile keyword?
- 11) Differences between synchronized and volatile keyword in Java?
- 12) Can you again start Thread?
- 13) What is race condition in multithreading and how can we solve it?
- 14) How threads communicate between each other?
- 15) Why wait(), notify() and notifyAll() are in Object class and not in Thread class?
- 16) Is it important to acquire object lock before calling wait(), notify() and notifyAll()?
- 17) How can you solve consumer producer problem by using wait() and notify() method?
- 18) How to solve Consumer Producer problem without using wait() and notify() methods, where consumer can consume only when production is over.?
- 19) Difference between wait() and wait(long timeout), What are thread states when these method are called?
- 20) How can you solve consumer producer pattern by using BlockingQueue?
- 21) What is deadlock in multithreading? Write a program to form DeadLock in multi threading and also how to solve DeadLock situation. What measures you should take to avoid deadlock?
- 22) Have you ever generated thread dumps or analyzed Thread Dumps?
- 23) What is life cycle of Thread, explain thread states?
- 24) Are you aware of preemptive scheduling and time slicing?
- 25) What are daemon threads?
- 26) Can a constructor be synchronized?
- 27) Can you find whether thread holds lock on object or not?
- 28) As stop() method is deprecated, How can we terminate or stop infinitely running thread in java?
- 29) what is significance of yield() method, what state does it put thread in?
- 30) What is significance of sleep() method in detail, what statedoes it put thread in ?
- 31) Difference between wait() and sleep() ?

- 32) Differences and similarities between yield() and sleep()?
- 33) What are thread priorities?
- 34) How thread can enter waiting, sleeping and blocked state and how can they go to runnable state ?
- 35) Difference between notify() and notifyAll() methods, can you write a code to prove your point?
- 36) Does thread leaves object lock when sleep() method is called?
- 37) Does thread leaves object lock when wait() method is called?
- 38) What will happen if we don't override run method?
- 39) What will happen if we override start method?
- 40) Can we acquire lock on class? What are ways in which you can acquire lock on class?
- 41) Difference between object lock and class lock?
- 42) Suppose you have 2 threads (Thread-1 and Thread-2) on same object. Thread-1 is in synchronized method1(), can Thread-2 enter synchronized method2() at same time?
- 43) Suppose you have 2 threads (Thread-1 and Thread-2) on same object. Thread-1 is in static synchronized method1(), can Thread-2 enter static synchronized method2() at same time?
- 44) Suppose you have 2 threads (Thread-1 and Thread-2) on same object. Thread-1 is in synchronized method1(), can Thread-2 enter static synchronized method2() at same time?
- 45) Suppose you have thread and it is in synchronized method and now can thread enter other synchronized method from that method?
- 46) Suppose you have thread and it is in static synchronized method and now can thread enter other static synchronized method from that method?
- 47) Suppose you have thread and it is in static synchronized method and now can thread enter other non static synchronized method from that method?
- 48) Suppose you have thread and it is in synchronized method and now can thread enter other static synchronized method from that method?
- 49) Suppose you have 2 threads (Thread-1 on object1 and Thread-2 on object2). Thread-1 is in synchronized method1(), can Thread-2 enter synchronized method2() at same time?
- 50) Suppose you have 2 threads (Thread-1 on object1 and Thread-2 on object2). Thread-1 is in static synchronized method1(), can Thread-2 enter static synchronized method2() at same time?

Week-6

Aim: Write a Java program for the following:

- i. Create a doubly linked list of elements.
- ii. Delete a given element from the above list.
- iii. Display the contents of the list after deletion.

Program:

```
import java.util.Scanner;
/* Class Node */
class Node
{
    protected int data;
    protected Node next, prev;
    /* Constructor */
    public Node()
    {
        next = null;
        prev = null;
        data = 0;
    }
    /* Constructor */
    public Node(int d, Node n, Node p)
    {
        data = d;
        next = n;
        prev = p;
    }
    /* Function to set link to next node */
    public void setLinkNext(Node n)
    {
        next = n;
    }
    /* Function to set link to previous node */
    public void setLinkPrev(Node p)
    {
        prev = p;
    }
    /* Funtion to get link to next node */
    public Node getLinkNext()
    {
        return next;
    }
    /* Function to get link to previous node */
```

```

public Node getLinkPrev()
{
    return prev;
}
/* Function to set data to node */
public void setData(int d)
{
    data = d;
}

/* Function to get data from node */
public int getData()
{
    return data;
}
}

/* Class linkedList */
class linkedList
{
    protected Node start;
    protected Node end ;
    public int size;
    /* Constructor */
    public linkedList()
    {
        start = null;
        end = null;
        size = 0;
    }
    /* Function to check if list is empty */
    public boolean isEmpty()
    {
        return start == null;
    }
    /* Function to get size of list */
    public int getSize()
    {
        return size;
    }
    /* Function to insert element at begining */
    public void insertAtStart(int val)
    {
        Node nptr = new Node(val, null, null);
        if(start == null)

```

```

        {
            start = nptr;
            end = start;
        }
        else
        {
            start.setLinkPrev(nptr);
            nptr.setLinkNext(start);
            start = nptr;
        }
        size++;
    }
    /* Function to insert element at end */
    public void insertAtEnd(int val)
    {
        Node nptr = new Node(val, null, null);
        if(start == null)
        {
            start = nptr;

            end = start;
        }
        else
        {
            nptr.setLinkPrev(end);
            end.setLinkNext(nptr);
            end = nptr;
        }
        size++;
    }
    /* Function to insert element at position */
    public void insertAtPos(int val , int pos)
    {
        Node nptr = new Node(val, null, null);
        if (pos == 1)
        {
            insertAtStart(val);
            return;
        }
        Node ptr = start;
        for (int i = 2; i <= size; i++)
        {
            if (i == pos)
            {
                Node tmp = ptr.getLinkNext();
                ptr.setLinkNext(nptr);
            }
        }
    }

```

```
        nptr.setLinkPrev(ptr);
        nptr.setLinkNext(tmp);
        tmp.setLinkPrev(nptr);
    }
    ptr = ptr.getLinkNext();
}
size++;
}
/* Function to delete node at position */
public void deleteAtPos(int pos)
{
    if (pos == 1)
    {
        if (size == 1)
        {
            start = null;
            end = null;
            size = 0;
            return;
        }
        start = start.getLinkNext();
        start.setLinkPrev(null);
        size--;
        return ;
    }
    if (pos == size)
    {
        end = end.getLinkPrev();
        end.setLinkNext(null);
        size-- ;
    }
    Node ptr = start.getLinkNext();
    for (int i = 2; i <= size; i++)
    {
        if (i == pos)
        {
            Node p = ptr.getLinkPrev();
            Node n = ptr.getLinkNext();
            p.setLinkNext(n);
            n.setLinkPrev(p);
            size-- ;
            return;
        }
        ptr = ptr.getLinkNext();
    }
}
```



```

/* Function to display status of list */
public void display()
{
    System.out.print("\nDoubly Linked List = ");
    if (size == 0)
    {
        System.out.print("empty\n");
        return;
    }
    if (start.getLinkNext() == null)
    {
        System.out.println(start.getData() );
        return;
    }
    Node ptr = start;
    System.out.print(start.getData()+ " <-> ");
    ptr = start.getLinkNext();
    while (ptr.getLinkNext() != null)
    {
        System.out.print(ptr.getData()+ " <-> ");
        ptr = ptr.getLinkNext();
    }
    System.out.print(ptr.getData()+ "\n");
}
}
/* Class DoublyLinkedList */
public class DoublyLinkedList
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        /* Creating object of linkedList */
        linkedList list = new linkedList();
        System.out.println("Doubly Linked List Test\n");
        char ch;
        /* Perform list operations */
        do

        {
            System.out.println("\nDoubly Linked List Operations\n");
            System.out.println("1. insert at begining");
            System.out.println("2. insert at end");
            System.out.println("3. insert at position");
            System.out.println("4. delete at position");
            System.out.println("5. check empty");
            System.out.println("6. get size");
        }
    }
}

```

```

        int choice = scan.nextInt();
    switch (choice)
    {
    case 1 :
        System.out.println("Enter integer element to insert");
        list.insertAtStart( scan.nextInt() );
        break;
    case 2 :
        System.out.println("Enter integer element to insert");
        list.insertAtEnd( scan.nextInt() );
        break;
    case 3 :
        System.out.println("Enter integer element to insert");
        int num = scan.nextInt() ;
        System.out.println("Enter position");
        int pos = scan.nextInt() ;
        if (pos < 1 || pos > list.getSize() )
            System.out.println("Invalid position\n");
        else
            list.insertAtPos(num, pos);
        break;
    case 4 :
        System.out.println("Enter position");
        int p = scan.nextInt() ;
        if (p < 1 || p > list.getSize() )
            System.out.println("Invalid position\n");
        else
            list.deleteAtPos(p);
        break;
    case 5 :
        System.out.println("Empty status = "+ list.isEmpty());
        break;
    case 6 :
        System.out.println("Size = "+ list.getSize() +" \n");
        break;
    default :
        System.out.println("Wrong Entry \n ");
        break;
    }
    /* Display List */
    list.display();
    System.out.println("\nDo you want to continue (Type y or n) \n");
    ch = scan.next().charAt(0);
    } while (ch == 'Y' || ch == 'y');
}
}

```

Output:

```

Command Prompt - java DoublyLinkedList
C:\Users\MLRITM\Desktop>java DoublyLinkedList
Doubly Linked List Test

Doubly Linked List Operations
1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
1
Enter integer element to insert
25
Doubly Linked List = 25
Do you want to continue <Type y or n>
y
Doubly Linked List Operations
1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
2
Enter integer element to insert
35
Doubly Linked List = 25 <-> 35
Do you want to continue <Type y or n>
y
Doubly Linked List Operations
1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
3
Enter integer element to insert
45
Enter position
2
Doubly Linked List = 25 <-> 45 <-> 35
    
```

```

C:\> Command Prompt - java DoublyLinkedList
Doubly Linked List = 25 <-> 45 <-> 35
Do you want to continue <Type y or n>
y
Doubly Linked List Operations
1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
4
Enter position
3
Doubly Linked List = 25 <-> 45
Do you want to continue <Type y or n>
y
Doubly Linked List Operations
1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
5
Empty status = false
Doubly Linked List = 25 <-> 45
Do you want to continue <Type y or n>
y
Doubly Linked List Operations
1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
6
Size = 2
Doubly Linked List = 25 <-> 45
    
```

```

C:\> Command Prompt
Do you want to continue <Type y or n>
n
C:\Users\MLRITM\Desktop>
    
```

Viva Questions:

1. What is a Linked list?
2. Can you represent a Linked list graphically?
3. How many types of Linked lists are there?
4. How to represent a linked list node?
5. Describe the steps to insert data at the starting of a singly linked list.
6. How to insert a node at the end of Linked list?
7. How to insert a node in random location in the list?
8. How to delete a node from linked list?
9. How to reverse a singly linked list?
10. How would you remove a node from a doubly linked list?
11. How to find if linked list has a loop?
12. What is difference between Singly Linked List and Doubly Linked List in Java?
13. How to find 3rd element from end in a linked list in one pass?
14. Which interfaces are implemented by Linked List in Java?
15. What is the package name for Linked List class in Java?
16. Difference between ArrayList and LinkedList ?
17. If you are given a choice to use either ArrayList and LinkedList, Which one would you use and Why ?
18. Is it legal to initialize List like this ?
19. What is the difference between ArrayList and LinkedList ?
20. Which is the Parent Class of LinkedList class
21. What type of memory allocation is referred for Linked lists?
22. Mention what is traversal in linked lists?
23. Describe what is Node in link list? And name the types of Linked Lists?
24. Mention what is the difference between Linear Array and Linked List?
25. Mention what are the applications of Linked Lists?
26. What does the dummy header in linked list contain?
27. Mention what is the difference between singly and doubly linked lists?
28. A linear collection of data elements where the linear node is given by means of pointer is called?
29. What are the various access specifiers for Java classes?
30. What's the purpose of Static methods and static variables?
31. What is data encapsulation and what's its significance?
32. What are Loops in Java? What are three types of loops?
33. What is an infinite Loop? How infinite loop is declared?
34. What is the difference between continue and break statement?
35. What is the difference between double and float variables in Java?
36. What is ternary operator? Give an example.
37. How can you generate random numbers in Java?
38. What is default switch case? Give example.
39. Can main() method in Java can return any data?
40. Can we declare a class as Abstract without having any abstract method?
41. What's the difference between an Abstract Class and Interface in Java?
42. Can we declare the main method of our class as private?

43. How can we pass argument to a function by reference instead of pass by value?
44. How an object is serialized in java?
45. When we should use serialization?
46. Is it compulsory for a Try Block to be followed by a Catch Block in Java for Exception handling?
47. Is there any way to skip Finally block of exception even if some exception occurs in the exception block?
48. When the constructor of a class is invoked?
49. Can we override static methods of a class?
50. How garbage collection is done in Java?

Week-7

Aim: 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.

Program:

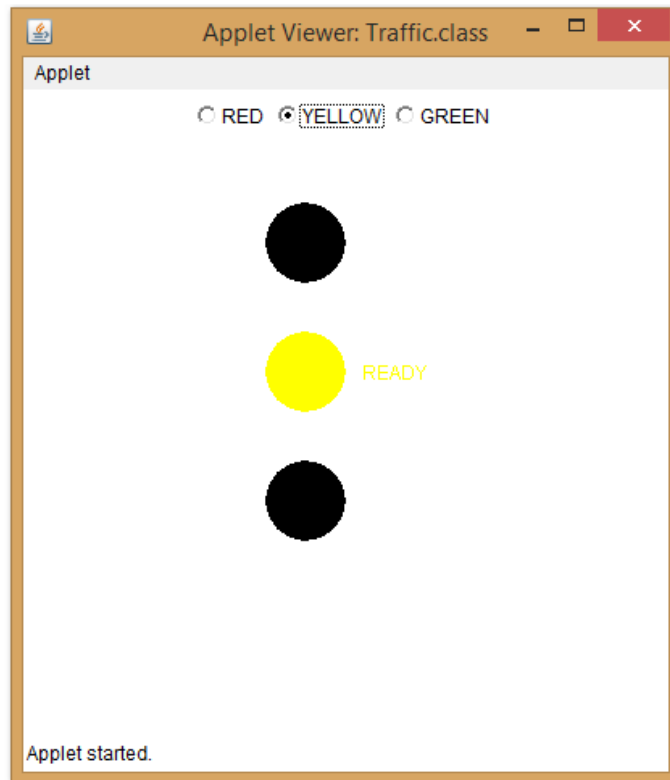
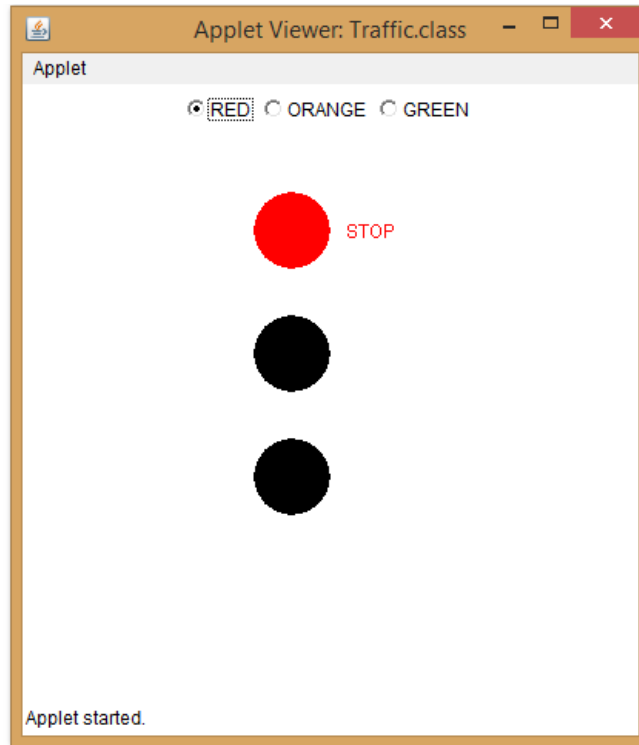
```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="Traffic.class" width=400 height=400></applet>*/

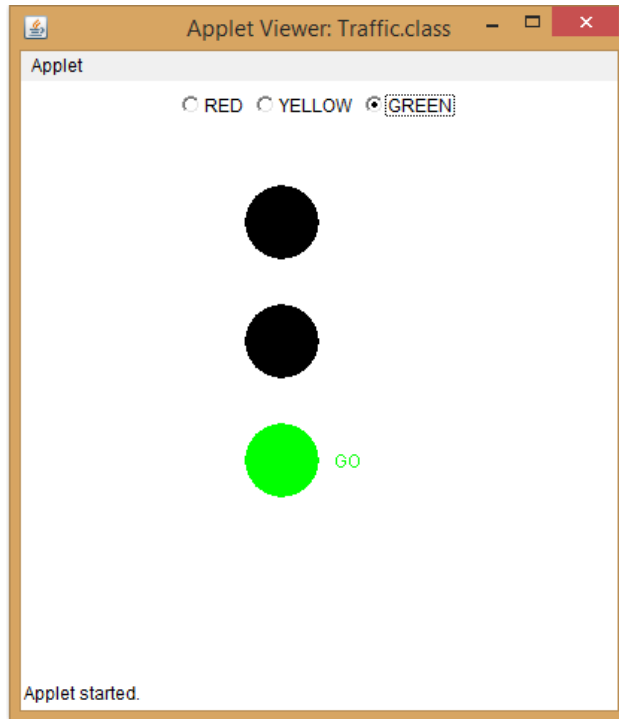
public class Traffic extends Applet implements ItemListener
{
    int colourNum; //global variable which is responsible for changing the light

    CheckboxGroup cbg;
    Checkbox red, yellow, green;
    String msg=" ";
public void init ()
{
    cbg=new CheckboxGroup();
    red=new Checkbox("RED",cbg,true);
    yellow=new Checkbox("YELLOW",cbg,true);
    green=new Checkbox("GREEN",cbg,true);
    add(red);
    add(yellow);
    add(green);
    red.addItemListener(this);
    yellow.addItemListener(this);
    green.addItemListener(this);
}

public void itemStateChanged(ItemEvent ie)
{
    if (ie.getSource () == red)
        colourNum = 1;
    else if (ie.getSource () == yellow)
        colourNum = 2;
    else
        colourNum = 3;
    repaint ();
}
}
```

```
public void paint (Graphics g) // responsible for graphics "within" the window
{
    g.setColor(Color.black);
    g.fillOval (150, 70, 50, 50); // red light
    g.fillOval (150, 150, 50, 50); // yellow light
    g.fillOval (150, 230, 50, 50); // green light
    switch (colourNum)
    {
        case 1: g.setColor (Color.red);
                g.fillOval (150,70,50,50); // red light
                msg="STOP";
                g.drawString(msg,210,100);
                break;
        case 2: g.setColor(Color.yellow);
                g.fillOval (150,150,50,50); // yellow light
                msg="READY";
                g.drawString(msg,210,180);
                break;
        case 3: g.setColor(Color.green);
                g.fillOval (150,230,50,50); // green light
                msg="GO";
                g.drawString(msg,210,260);
                break;
    }
}
}
```



Viva Questions:

1. What is AWT?
2. What is component?
3. How to interact with the Java system at runtime?
4. What is the super class of all components of Java?
5. What is a container?
6. What is the super class of all containers?
7. What is a layout manager?
8. How many layout managers are available in Java?
9. What is the style of arranging components in a container by FlowLayout manager?
10. How BorderLayout places the components?
11. What is the style of GridLayout?
12. When CardLayout can be used?
13. When GridBagLayout can be used?
14. What is a panel?
15. What are the default layout managers for containers?
16. Which layout manager gives the minimum size to a component?
17. What is the method used to place some text in the text field?
18. What is the method used to get the data entered by the user in the text field?
19. What for text field is used?
20. What is the difference between text field and text area?
21. What is the method used to change the characters entered by the user in the text field (used for password)?
22. How to make the text field non-editable by the user (user cannot enter anything)?
23. What is the method used to change the background color of components like text field?
24. What is the method used to change the foreground (text) color of components like text field?
25. What is the method used to know the label of the button clicked by the user?
26. What is the super class of TextField and TextArea?
27. What is the method used to change the text of a Label?
28. What is the method used to retrieve the text of a Label?
29. How many ways you can align the label in a container?
30. What is the method used to change the label of a button?
31. What is the method used to retrieve the label of a button?
32. What is HeadlessException?
33. What is the listener used to handle the events of a text field?
34. What are the component and container class?
35. What is the parameter specification for the public static void main method?
36. What is the difference between the paint() and repaint() method?
37. What interface is extended by AWT event listener?

38. How are the elements of different layouts organized?
39. What is the default layout for Applet?
40. What is the difference between Grid and GridbagLayout?
41. What is a layout manager?
42. What is paint method?
43. What is the purpose of repaint method?
44. Which containers use a Border layout as their default layout?
45. What is the difference between choice and list?
46. What is the difference between a window and a frame?
47. What are the subclasses of the Container class?
48. Which method is method to set the layout of a container?
49. Which method will cause a Frame to be displayed?
50. What are the subclass of TextComponent class?

Week-8

Aim: 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.

Program:

```
abstract class Shape
{
    int height;
    int width;
    int radius;
    abstract int printArea();
}

class Rectangle extends Shape
{
    Rectangle(int width,int height)
    {
        this.width=width;
        this.height=height;
    }
    int printArea()
    {
        System.out.println("Inside Area for Rectangle.");
        return height * width;
    }
}

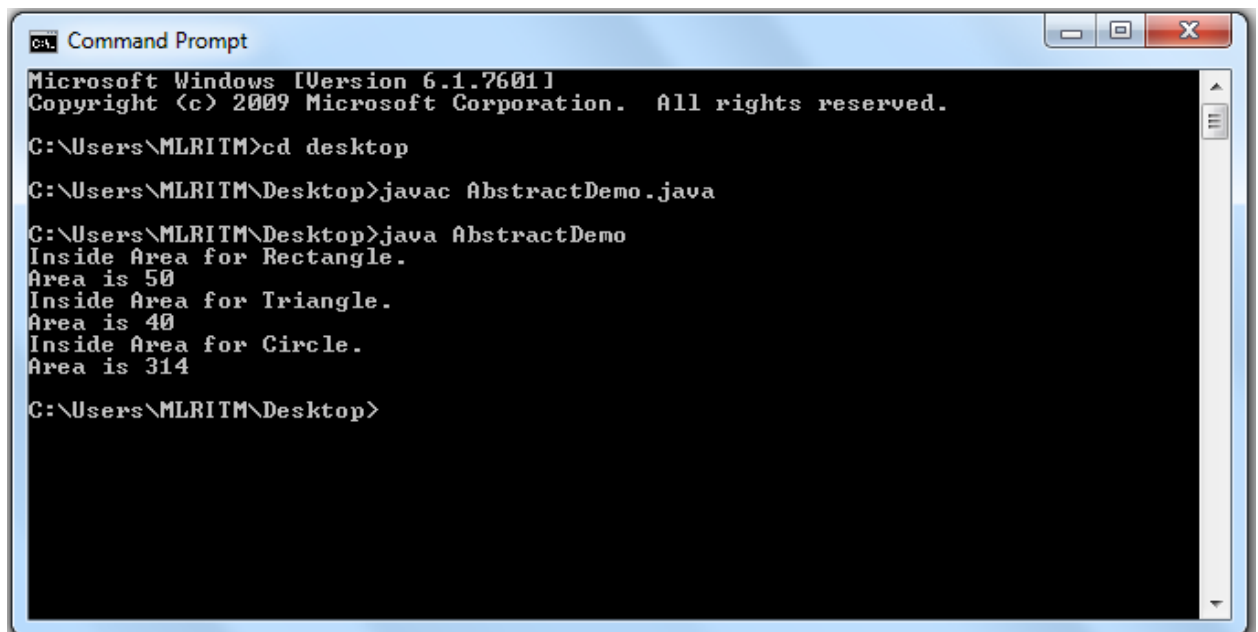
class Triangle extends Shape
{
    Triangle(int width, int height)
    {
        this.width=width;
        this.height=height;
    }
    int printArea()
    {
        System.out.println("Inside Area for Triangle.");
        return height * width / 2;
    }
}

class Circle extends Shape
{
    Circle(int r)
```

```
{
    radius=r;
}
int printArea()
{
    System.out.println("Inside Area for Circle.");
    int area=(int)(3.14*radius*radius);
    return(area);
}
}
public class AbstractDemo
{
    public static void main(String args[])
    {
        Rectangle r = new Rectangle(10, 5);
        Triangle t = new Triangle(10, 8);
        Circle c=new Circle(10);

        System.out.println("Area is " +r.printArea());
        System.out.println("Area is " + t.printArea());
        System.out.println("Area is " +c.printArea());

    }
}
```

Output:

```
Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\MLRITM>cd desktop
C:\Users\MLRITM\Desktop>javac AbstractDemo.java
C:\Users\MLRITM\Desktop>java AbstractDemo
Inside Area for Rectangle.
Area is 50
Inside Area for Triangle.
Area is 40
Inside Area for Circle.
Area is 314
C:\Users\MLRITM\Desktop>
```

Viva Questions:

1. What is abstraction in java?
2. How to achieve abstraction in java?
3. What is abstract class in java?
4. Can we create instance of abstract class?
5. Can we define abstract class without abstract method?
6. Can we declare abstract method in non-abstract class?
7. What is interface in java?
8. Why we use interface in java?
9. Can we create instance of interface?
10. Can we declare abstract method as static?
11. Can we declare abstract method as final?
12. Can we declare abstract method as private?
13. Can we use public, protected and default modifiers with abstract method?
14. Can we declare local inner class as abstract?
15. Method in interface are by default public and abstract. true or false?
16. Data member in interface are by default public, static, and final. true or false?
17. Can abstract class implements interface in java?
18. Can we use abstract keyword with constructor?
19. Abstract classes can be nested. true or false?
20. Can abstract class have constructor in java?
21. Difference between abstraction and encapsulation in java?
22. What is the difference between abstract class and interface?
23. Can abstract method declaration include throws clause?
24. What will happen if we do not override all the abstract methods in sub-class?
25. How can we define an abstract class?
26. How to declare an abstract method?
27. Can we define abstract class without abstract method?
28. Can we create object for abstract class?
29. Is it possible to declare abstract method as static?
30. Can we declare abstract method as final?
31. Is it possible to declare abstract method as public ?
32. Is it possible to declare abstract method with default?
33. Is it possible to declare abstract method with protected modifier?
34. What are the valid and invalid keywords or modifier with abstract class?
35. Can abstract method declaration include throws clause?
36. What happens if sub class not overriding abstract methods?
37. Can we escape of overriding abstract class in sub class which is extending abstract class?
38. What does it mean that a method or class is abstract?
39. What must a class do to implement an interface?
40. What is interface? How to support multiple inheritance in Java?
41. Can you make an instance of an abstract class?
42. If interface & abstract class have same methods and those methods contain no implementation, which one would you prefer?
43. Why the main method is static in java?
44. What happens if you remove static modifier from the main method?

45. What is “this” keyword in java?
46. How abstract classes are similar or different in Java from C++?
47. Can we overload main() method?
48. How is inheritance in C++ different from Java?
49. Can we override private methods in Java?
50. Why method overloading is not possible by changing the return type in java?

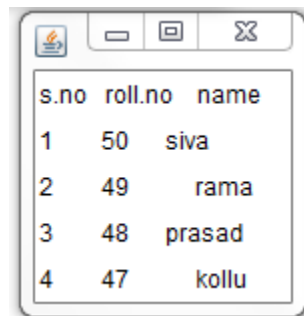
Week-9

Aim: 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 to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.

Program:

```
import java.io.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
class A extends Frame
{
public A()
{
setSize(600, 600);
GridLayout g = new GridLayout(0, 4);
setLayout(g);
try
{
FileInputStream fin = new FileInputStream("");
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 Label(i));
}
}
}
catch (Exception ex)
{
}
pack();
setVisible(true);
}
```

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

Output:

s.no	roll.no	name
1	50	siva
2	49	rama
3	48	prasad
4	47	kollu

Viva Questions:

1. How to remove duplicates from ArrayList in Java?
2. How to reverse ArrayList in Java?
3. Difference between an array and ArrayList in Java?
4. How to synchronize ArrayList in Java?
5. When to use ArrayList and LinkedList in Java?
6. Difference between ArrayList and HashSet in Java?
7. How to loop over ArrayList in Java?
8. Difference between Vector and ArrayList in Java?
9. How to create and initialize ArrayList in one line?
10. How to sort ArrayList in Java?
11. Difference between HashMap and ArrayList in Java?
12. How to use ArrayList in Java?
13. How to convert ArrayList to String in Java?
14. How to get sublist from ArrayList in Java?
15. Difference between length() of array and size() of ArrayList in Java?
16. How to remove objects from ArrayList in Java?
17. How to make ArrayList read only in Java?
18. Tell something about BufferedWriter ?
19. What are flush() and close() used for ?
20. What is Scanner class used for ?
21. What is a stream and what are the types of Streams and classes of the Streams?
22. What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?
23. Which is the abstract parent class of FileWriter ?
24. Which class is used to read streams of raw bytes from a file?
25. Which is the Parent class of FileInputStream ?
26. Which exceptions should be handled with the following code ?
27. Which is the Parent Class of ByteArrayInputStream and AudioInputStream ?
28. Which interfaces are implemented by InputStream?
29. What is the package name for ObjectInputStream class?
30. Which is the Parent Class of StringBufferInputStream class?
31. Which is the Parent Class of SequenceInputStream class?
32. Which is the Parent Class of FilterInputStream class?
33. What is the package name for FilterInputStream class?
34. Which interfaces are implemented by FileInputStream?
35. What Is A Layout Manager?
36. Which Method Is Method To Set The Layout Of a Container?
37. What Are The Default Layouts For an Applet, Frame and Panel?
38. Which Method Will Cause a Frame To Be Displayed?
39. In which package GridLayout class is present?
40. Which interfaces was implemented by GridLayout?
41. What is the use of GridLayout()?
42. What is the use of GridLayout(int rows, int cols)

43. What is the use of GridLayout(int rows, int cols, int hgap, int vgap)
44. What is method used to find number of rows in GridLayout?
45. What is method used to find number of columns in GridLayout?
46. What is method used to find vertical gap between the components in GridLayout?
47. How to set number of columns in GridLayout?
48. How to set number of rows in GridLayout?
49. How to set vertical gap between the components in GridLayout?
50. How to set horizontal gap between the components in GridLayout?

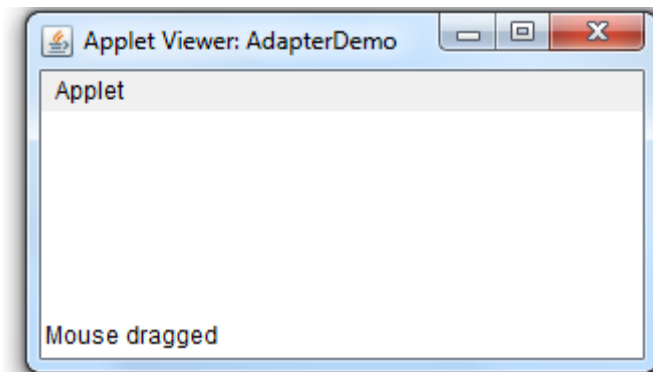
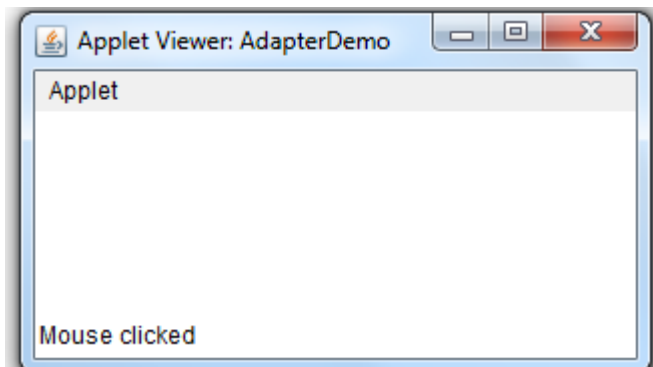
Week-10

Aim: 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).

Program:

```
// Demonstrate an adapter.
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/* <applet code="AdapterDemo" width=300 height=100>
</applet>
*/
public class AdapterDemo extends Applet {
    public void init() {
        addMouseListener(new MyMouseAdapter(this));
        addMouseMotionListener(new MyMouseMotionAdapter(this));
    }
}
class MyMouseAdapter extends MouseAdapter {
    AdapterDemo adapterDemo;
    public MyMouseAdapter(AdapterDemo adapterDemo) {
        this.adapterDemo = adapterDemo;
    }
    // Handle mouse clicked.
    public void mouseClicked(MouseEvent me) {
        adapterDemo.showStatus("Mouse clicked");
    }
}
class MyMouseMotionAdapter extends MouseMotionAdapter {
    AdapterDemo adapterDemo;
    public MyMouseMotionAdapter(AdapterDemo adapterDemo) {
        this.adapterDemo = adapterDemo;
    }
    // Handle mouse dragged.
    public void mouseDragged(MouseEvent me) {
        adapterDemo.showStatus("Mouse dragged");
    }
}
```

Output:



Viva Questions:

1. What is the purpose of Event object?
2. What is an action event?
3. What are the different kinds of event listeners?
4. What is event adapter class?
5. When should we use an event adapter class?
6. What is the relationship between an event-listener interface and an event-adapter class?
7. What is the difference between the paint() and repaint() methods?
8. What is the Event handling?
9. What is Adapter class?
10. What interface is extended by AWT event listener?
11. What is an event and what are the models available for event handling?
12. What are the advantages of the event delegation model over the event inheritance model?
13. What event results from the clicking of a button?
14. What is source?
15. What is a listener?
16. What is the highest level event class of the event delegation model?
17. How key events can be handled?
18. How mouse events can be handled?
19. What is an event?
20. How to add listener?
21. How to remove listener?
22. What is the super class of all the event classes?
23. Which package contains all the events?
24. What are the methods of EventObject class?
25. What is the super class of all the awt event classes?
26. What is the use of ActionEvent?
27. Which event is generated when button is clicked?
28. Which event is generated when scroll bar is manipulated?
29. Which event is generated when a component hidden?
30. Which event is generated when component is added or hidden?
31. Which event is generated when component gains or losses keyboard focus?
32. What is the abstract super class of all input events?
33. Which event is generated when input is received from keyboard?
34. Which event is generated when the mouse wheel is moved?
35. Which event is generated when the text are is changed?
36. What are the methods of MouseListener interface?
37. What are the methods of MouseMotionListener interface?
38. What are the methods of MouseWheelListener interface?
39. What is the adapter class used to handle mouse events?

40. What is the adapter class used to handle key events?
41. What is the source to generate action events?
42. What is the source to generate item events?
43. What is the interface used for action events?
44. What is the interface used for adjustment events?
45. Write down the syntax of addActionListener() method?
46. Write down the syntax of removeActionListener() method?
47. What is the use of MouseEvent class?
48. What is the use of KeyEvent class?
49. What is the use of getX() and getY() methods?
50. What is the use of MouseWheelEvent class?

Week-11

Aim: 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).

Program:

```
import java.util.*;
import java.io.*;
public class PhoneDictionary
{
    public static void main(String[] args)
    {
        try
        {
            FileInputStream fs = new FileInputStream("E:\\siva\\phone.txt");
            Scanner sc = new Scanner(fs).useDelimiter("\\s+");
            Hashtable<String, String> ht = new Hashtable<String, String>();
            String[] arrayList;
            String a;
            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("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();
            }
        }
    }
}
```

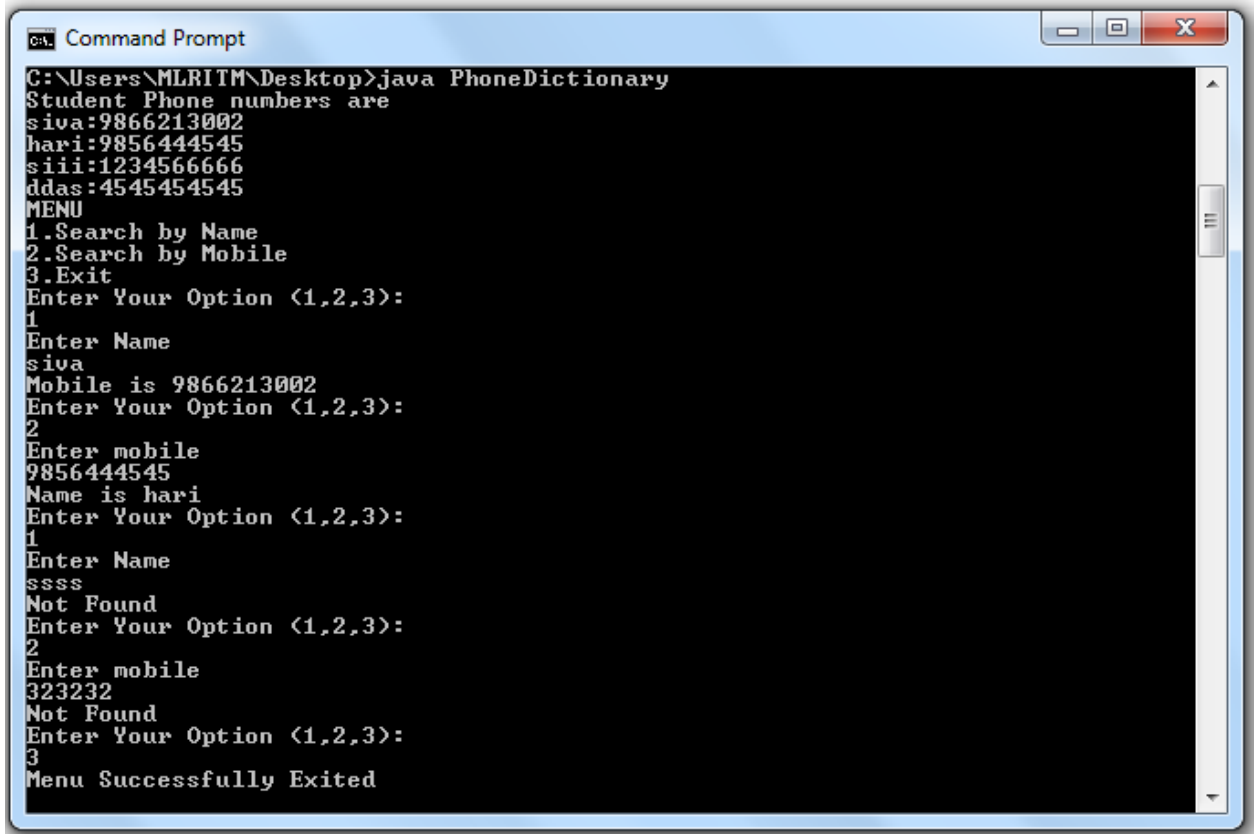
```
switch (opt)
{
    case "1":    System.out.println("Enter Name");
                name = s.next();
                if (ht.containsKey(name))
                {
                    System.out.println("Mobile is " + ht.get(name));
                }
                else
                {
                    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());
    }
}
}

```

Output:



Viva Questions:

1. What are Collection related features in Java 8?
2. What is Java Collections Framework? List out some benefits of Collections framework?
3. What is the benefit of Generics in Collections Framework?
4. What are the basic interfaces of Java Collections Framework?
5. Why Collection doesn't extend Cloneable and Serializable interfaces?
6. Why Map interface doesn't extend Collection interface?
7. What is an Iterator?
8. What is difference between Enumeration and Iterator interface?
9. Why there is not method like Iterator.add() to add elements to the collection?
10. Why Iterator don't have a method to get next element directly without moving the cursor?
11. What is different between Iterator and ListIterator?
12. What are different ways to iterate over a list?
13. What do you understand by iterator fail-fast property?
14. What is difference between fail-fast and fail-safe?
15. How to avoid ConcurrentModificationException while iterating a collection?
16. Why there are no concrete implementations of Iterator interface?
17. What is UnsupportedOperationException?
18. How HashMap works in Java?
19. What is the importance of hashCode() and equals() methods?
20. Can we use any class as Map key?
21. What are different Collection views provided by Map interface?
22. What is difference between HashMap and Hashtable?
23. How to decide between HashMap and TreeMap?
24. What are similarities and difference between ArrayList and Vector?
25. What is difference between Array and ArrayList? When will you use Array over ArrayList?
26. What is difference between ArrayList and LinkedList?
27. Which collection classes provide random access of it's elements?
28. What is EnumSet?
29. Which collection classes are thread-safe?
30. What are concurrent Collection Classes?
31. What is BlockingQueue?
32. What is Queue and Stack, list their differences?
33. What is Collections Class?
34. What is Comparable and Comparator interface?
35. What is difference between Comparable and Comparator interface?
36. How can we sort a list of Objects?
37. While passing a Collection as argument to a function, how can we make sure the function will not be able to modify it?
38. How can we create a synchronized collection from given collection?
39. What are common algorithms implemented in Collections Framework?
40. What is Big-O notation? Give some examples?

41. What are best practices related to Java Collections Framework?
42. What is Java Priority Queue?
43. Why can't we write code as `List<Number> numbers = new ArrayList<Integer>();`?
44. Why can't we create generic array? or write code as `List<Integer>[] array = new ArrayList<Integer>[10];`
45. What is the root interface in collection hierarchy ?
46. What are the classes implementing List and Set interface ?
47. Which methods you need to override to use any object as key in HashMap ?
48. How to reverse the List in Collections ?
49. What is the difference between HashMap and Hashtable ?
50. What are Comparable and Comparator interfaces ? List the difference between them ?

Week-12

Aim: Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.

Program:

```
class Q
{
    int n;
    boolean valueSet = false;
    synchronized int get()
    {
        while(!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)
    {
        while(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
```

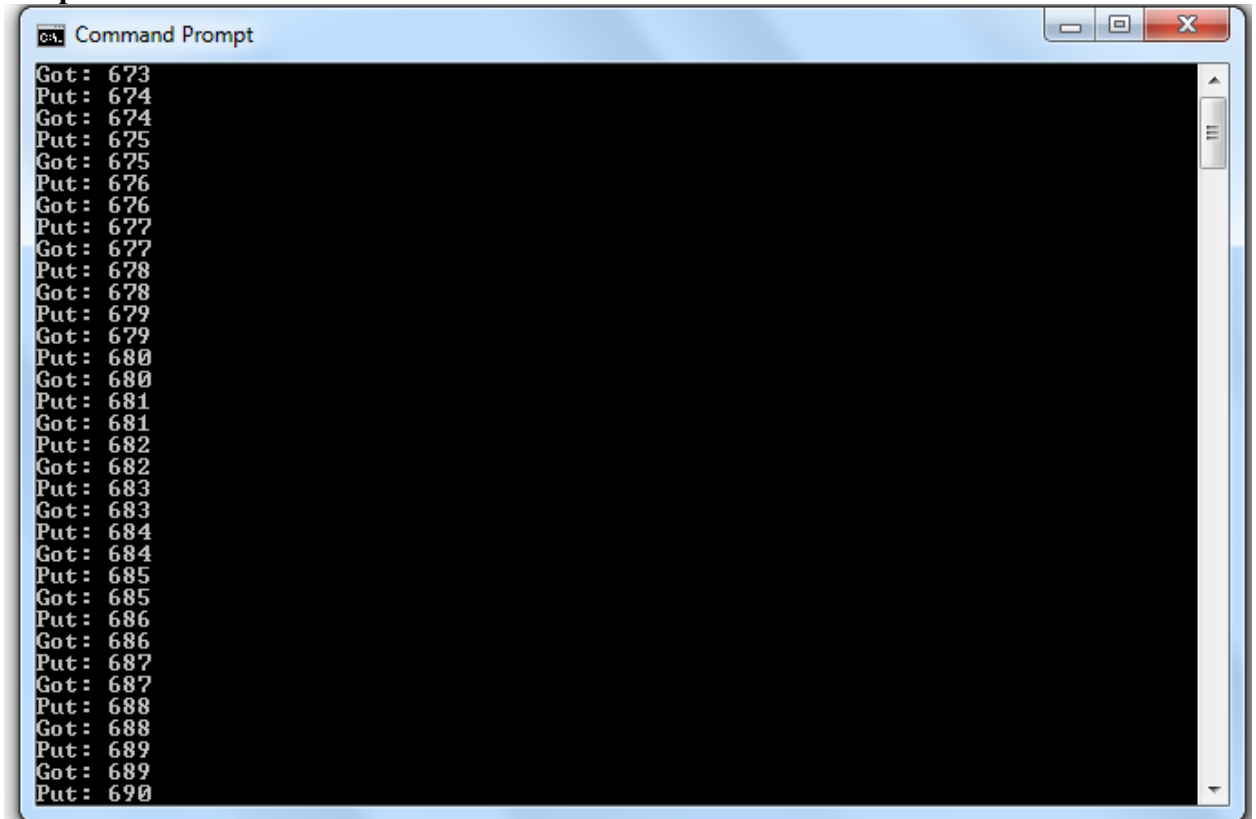
```

    {
        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 PCFixed
{
    public static void main(String args[])
    {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
    }
}

```

}

Output:



Viva Questions:

1. What is Thread in Java?
2. What is the difference between Thread and Process in Java?
3. How do you implement Thread in Java?
4. When to use Runnable vs Thread in Java?
5. What is the difference between start() and run() method of Thread class?
6. What is the difference between Runnable and Callable in Java?
7. What is the difference between CyclicBarrier and CountdownLatch in Java?
8. What is Java Memory model?
9. What is volatile variable in Java?
10. What is thread-safety? is Vector a thread-safe class?
11. What is race condition in Java? Given one example?
12. How to stop a thread in Java?
13. What happens when an Exception occurs in a thread?
14. How do you share data between two thread in Java?
15. What is the difference between notify and notifyAll in Java?
16. Why wait, notify and notifyAll are not inside thread class?
17. What is ThreadLocal variable in Java?
18. What is FutureTask in Java?
19. What is the difference between the interrupted() and isInterrupted() method in Java?
20. Why wait and notify method are called from synchronized block?
21. Why should you check condition for waiting in a loop?
22. What is the difference between synchronized and concurrent collection in Java?
23. What is the difference between Stack and Heap in Java?
24. What is thread pool? Why should you thread pool in Java?
25. Write code to solve Producer Consumer problem in Java?
26. How do you avoid deadlock in Java? Write Code?
27. What is the difference between livelock and deadlock in Java?
28. How do you check if a Thread holds a lock or not?
29. How do you take thread dump in Java?
30. Which JVM parameter is used to control stack size of a thread?
31. What is the difference between synchronized and ReentrantLock in Java?
32. There are three threads T1, T2, and T3? How do you ensure sequence T1, T2, T3 in Java?
33. What does yield method of Thread class do?
34. What is the concurrency level of ConcurrentHashMap in Java?
35. What is Semaphore in Java?
36. What happens if you submit a task when the queue of the thread pool is already filled?
37. What is the difference between the submit() and execute() method thread pool in Java?
38. What is blocking method in Java?
39. Is Swing thread-safe? What do you mean by Swing thread-safe?
40. What is the difference between invokeAndWait and invokeLater in Java?

41. Which method of Swing API are thread-safe in Java?
42. How to create an Immutable object in Java?
43. What is ReadWriteLock in Java?
44. What is busy spin in multi-threading?
45. What is the difference between the volatile and atomic variable in Java?
46. What happens if a thread throws an Exception inside synchronized block?
47. What is double checked locking of Singleton?
48. How to create thread-safe Singleton in Java?
49. List down 3 multi-threading best practice you follow?
50. How do you force to start a Thread in Java?

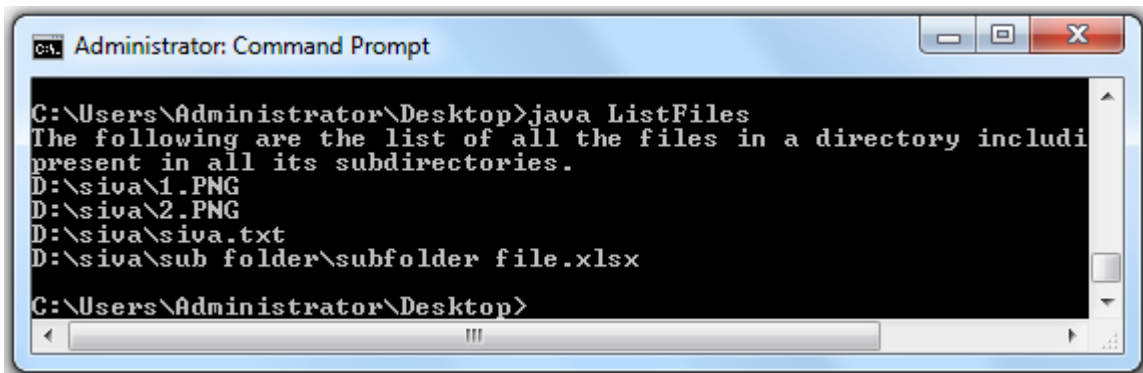
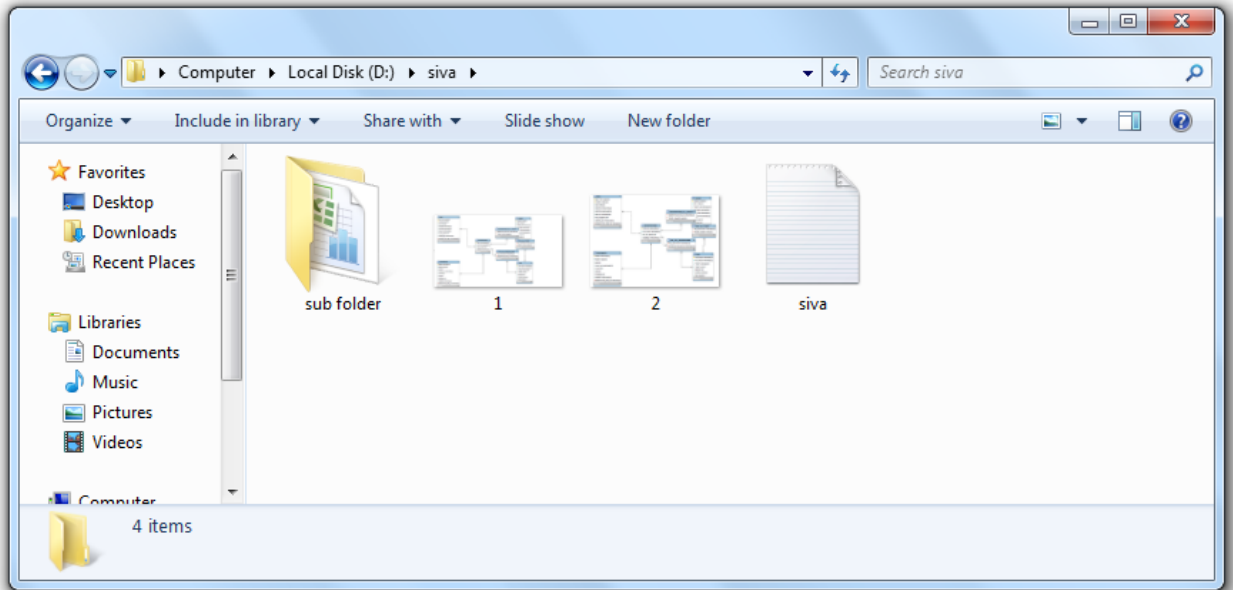
Week-13

Aim: Write a Java program to list all the files in a directory including the files present in all its subdirectories.

Program:

```
import java.io.File;
public class ListFiles
{
    public void listFilesAndFilesSubDirectories(String directoryName)
    {
        File directory = new File(directoryName);
        File[] fList = directory.listFiles();
        for (File file : fList)
        {
            if (file.isFile())
            {
                System.out.println(file.getAbsolutePath());
            }
            else if (file.isDirectory())
            {
                listFilesAndFilesSubDirectories(file.getAbsolutePath());
            }
        }
    }
    public static void main (String[] args)
    {
        ListFiles l= new ListFiles();
        final String directoryWindows ="D:\\siva";
        System.out.println("The following are the list of all the files in a directory including
the files present in all its subdirectories.");
        l.listFilesAndFilesSubDirectories(directoryWindows);
    }
}
```

Output:



Viva Questions:

1. How to Create New File in Java?
2. How to delete a File in Java?
3. What is Java File separator?
4. What is separatorChar?
5. What is pathSeparator?
6. What is pathSeparatorChar?
7. How to delete non empty directory in Java?
8. How to Rename a file?
9. How to Move a file in Java?
10. How to know File Size in Java?
11. How to Get File Extension in Java?
12. What is the method used to Check File Exists in Java?
13. How to check if File is Directory?
14. What is the method used to know File Last Modified Date in Java?
15. What is meant by FileNameFilter ?
16. What is the method used to know the File Path?
17. What is the method used to know thAbsolute Path ?
18. How to set File Permissions in Java ?
19. How to Copy a File in Java?
20. How to Open File in Java?
21. How to Read File in Java?
22. How to Read a File to String in Java?
23. How to read file line by line?
24. What is a file?
25. How to Append Text to File in Java?
26. What is InputStream?
27. What is the use of Random Access File?
28. How to write Object to File in Java?
29. How to read Object from File in Java?
30. How to Compile and Run a Java Program in another Java program?
31. How to Compress File or Directory using Java ZIP?
32. How to Unzip File?
33. What is Byte Stream?
34. List different types of streams?
35. What is character stream?
36. What is the use of OutputStream class?
37. Is Outputstream is an abstract class?
38. What is the use of Console class?
39. What is the use of Closeable interface?
40. Which exception is thrown by read()?
41. Which exception is thrown by readLine()?
42. What is the use of InputStreamReader class?

43. What is meant by System.err?
44. What is meant by System.out?
45. What is meant by System.in?
46. What is default device attached to System.err and System.out?
47. What is default device attached to System.in?
48. List out few methods of File class?
49. Which is the abstract parent class of FileWriter ?
50. Which class is used to read streams of characters from a file?

Week-14

Aim: Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order

Program:

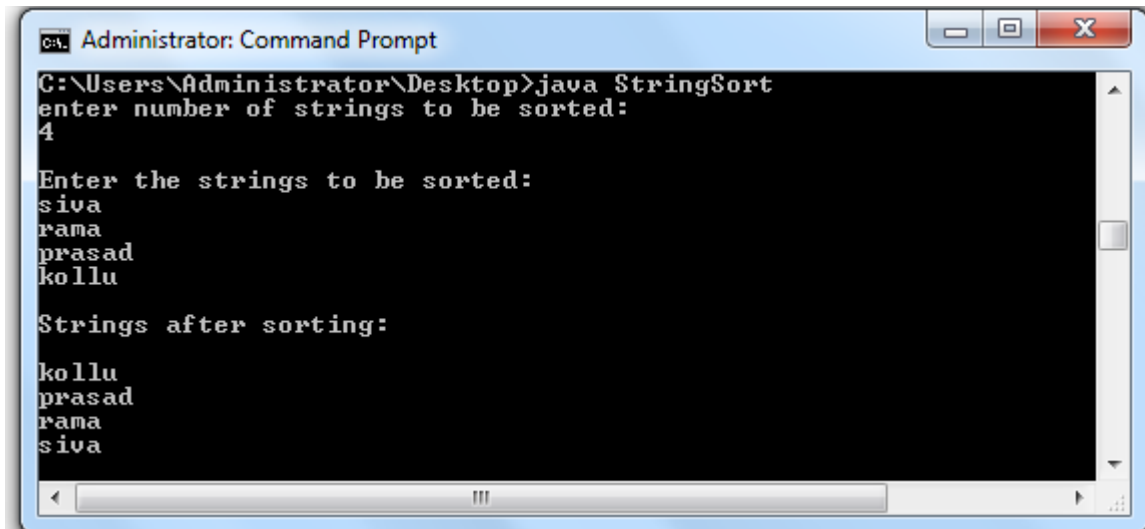
```
import java.util.Scanner;
public class StringSort
{
    String names[];
    int length;
    void sort(String array[])
    {
        if (array == null || array.length == 0)
        {
            return;
        }
        this.names = array;
        this.length = array.length;
        quickSort(0, length - 1);
    }
    void quickSort(int lowerIndex, int higherIndex)
    {
        int i = lowerIndex;
        int j = higherIndex;
        String pivot = this.names[lowerIndex + (higherIndex - lowerIndex) / 2];
        while (i <= j)
        {
            while (this.names[i].compareToIgnoreCase(pivot) < 0)
            {
                i++;
            }
            while (this.names[j].compareToIgnoreCase(pivot) > 0)
            {
                j--;
            }
            if (i <= j)
            {
                exchangeNames(i, j);
                i++;
            }
        }
    }
}
```

```

        j--;
    }
}
//call quickSort recursively
if (lowerIndex < j)
{
    quickSort(lowerIndex, j);
}
if (i < higherIndex)
{
    quickSort(i, higherIndex);
}
}
void exchangeNames(int i, int j)
{
    String temp = this.names[i];
    this.names[i] = this.names[j];
    this.names[j] = temp;
}

public static void main(String[] args)
{
    StringSort sorter= new StringSort();
    int n;
    System.out.println("enter number of strings to be sorted:");
    Scanner sc=new Scanner(System.in);
    n=sc.nextInt();
    String words[] =new String[n];
    System.out.println("\nEnter the strings to be sorted:");
    for(int i=0;i<n;i++)
        words[i]=sc.next();
    System.out.println("\nStrings after sorting:\n");
    sorter.sort(words);
    for (String i : words)
    {
        System.out.println(i);
    }
    System.out.println("");
}
}

```


Output:

```
Administrator: Command Prompt
C:\Users\Administrator\Desktop>java StringSort
enter number of strings to be sorted:
4

Enter the strings to be sorted:
siva
rama
prasad
kollu

Strings after sorting:
kollu
prasad
rama
siva
```

Viva Questions:

1. What is String in Java? String is a data type?
2. What are different ways to create String Object?
3. Write a method to check if input String is Palindrome?
4. Write a method that will remove given character from the String?
5. How can we make String upper case or lower case?
6. What is String subSequence method?
7. How to compare two Strings in java program?
8. How to convert String to char and vice versa?
9. How to convert String to byte array and vice versa?
10. Can we use String in switch case?
11. Write a program to print all permutations of String?
12. Write a function to find out longest palindrome in a given string?
13. Difference between String, StringBuffer and StringBuilder?
14. Why String is immutable or final in Java
15. How to Split String in java?
16. Why Char array is preferred over String for storing password?
17. How do you check if two Strings are equal in Java?
18. What is String Pool?
19. What does String intern() method do?
20. Does String is thread-safe in Java?
21. Why String is popular HashMap key in Java?
22. Is *String* a keyword in java?
23. Is *String* a primitive type or derived type?
24. In how many ways you can create string objects in java?
25. What is string constant pool?
26. What do you mean by mutable and immutable objects?
27. Which is the final class in these three classes – String, StringBuffer and StringBuilder?
28. How do you create mutable string objects?
29. Which one will you prefer among “==” and equals() method to compare two string objects?
30. How do you convert given string to char array?
31. Where exactly string constant pool is located in the memory?
32. What is the main difference between Java strings and C, C++ strings?
33. Can we call String class methods using string literals?
34. How do you remove all white spaces from a string in java?
35. How do you count the number of occurrences of each character in a string?
36. How do you find duplicate characters in a string?
37. How do you convert string to integer and integer to string in java?
38. How do you convert string to integer and integer to string in java?
39. How to check if String has all unique characters in java?
40. How to check if one String is rotation of another String in java?
41. What are the interfaces implemented by String class?
42. How to find the length of the string?
43. How to compare two string by ignoring cases?

44. How to check whether the string is empty or not?
45. How to find the index of a character in a string?
46. What is the method used to convert a string into lower case?
47. What is the method used to convert a string into upper case?
48. What is the use of trim() method?
49. How to replace a character in a string by another character?
50. What is the method used to concat strings?

Week-15

Aim: Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.

Program:

```
import java.util.Arrays;

import java.util.Scanner;

public class BubbleSort {

    // Bubble Sort Algorithm in Descending Order

    static int count=0;

    public static int[] Desc(int[] Arr) {

        int temp;

        for (int i = 0; i < Arr.length - 1; i++) {

            for (int j = 1; j < Arr.length - i; j++) {

                if (Arr[j - 1] < Arr[j]) {

                    temp = Arr[j - 1];

                    Arr[j - 1] = Arr[j];

                    Arr[j] = temp;

                }

            }

            System.out.println("Iteration " + (i + 1) + ": " + Arrays.toString(Arr));

            count++;

        }

        return Arr;

    }

}
```

```

public static void main(String[] args) {

System.out.print("enter the number of elements to be sorted:");

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int Arr[]=new int[n];

int i;

System.out.println("enter the elements");

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

Arr[i]=sc.nextInt();

System.out.println("Let's get started on Bubble Sort implementation in Java \n");

System.out.println("\n===== Descending Order
result:===== \n\n" + Arrays.toString(Desc(Arr)) + "\n");

System.out.println("Total number of iterations are:"+count);

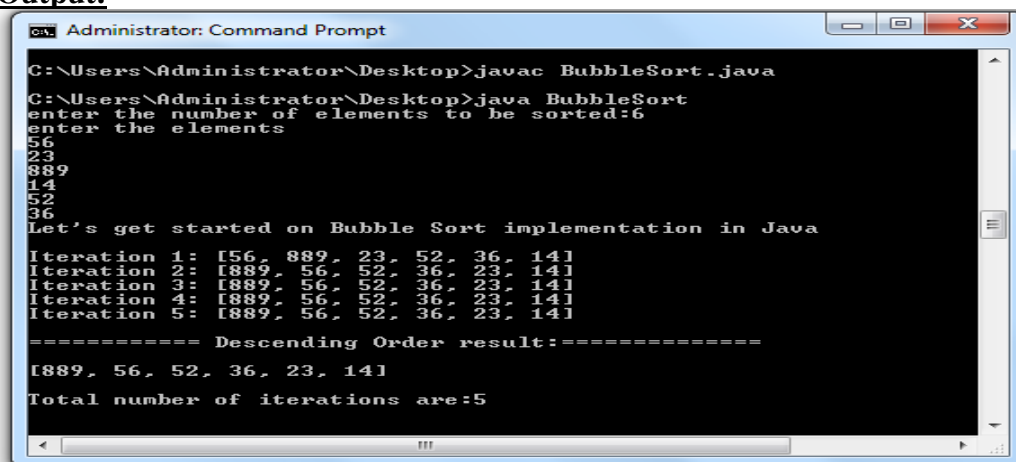
System.out.println("");

}

}

```

Output:



Viva Questions:

1. What do you mean by an Array?
2. How to create an Array?
3. What are the advantages and disadvantages of Array?
4. Can we change the size of an array at run time?
5. Can you declare an array without assigning the size of an array?
6. What is the default value of Array?
7. How to print element of Array?
8. How to compare Two Arrays?
9. How to sort an Array?
10. Can we declare array size as a negative number?
11. When will we get ArrayStoreException?
12. Can we add or delete an element after assigning an array?
13. What is the meaning of anonymous array? Explain with an example?
14. Is there any difference between `int[] a` and `int a[]`?
15. There are 2 int type array data type. One is containing 50 elements, and another one is containing 30 elements. Can we assign the array of 50 elements to an array of 30 elements?
16. `int a[] = new int[3]{1, 2, 3}` – is it a right way to declare arrays in java?
17. How to copy an array into another array?
18. What are “jagged” arrays in java?
19. When `ArrayIndexOutOfBoundsException` occurs?
20. Can you explain different steps of declaring multidimensional arrays in Java?
21. How do we search a specific element in an array?
22. If you do not initialize an array what will happen?
23. How do we find duplicate elements in an array?
24. Can we use Generics with the array?
25. How to iterate an array in java?
26. Where is the memory allocated for arrays in Java?
27. Can you tell me the class name of an array in Java?
28. “`int a[] = new int[3]{1, 2, 3}`” – This a legal way of defining the arrays?
29. What is the two-dimensional array?
30. Do we have 3-dimensional arrays in Java?
31. Can we make array volatile in Java?
32. Can you tell me the differences between Array and ArrayList?
33. We know that Arrays are objects so why cannot we write `strArray.length()`?
34. How to check array contains value or not?
35. How to get largest and smallest number in an array?
36. How to do the intersection of two sorted arrays?
37. How to get top two numbers from an array?
38. How to find the length an array?
39. What is the difference between `length()` and `length`?

40. What is the use of Scanner class?
41. List the methods of Scanner class?
42. What is the use of toString() method?
43. What is the use of nextInt() method?
44. Can you store String in an array of Integer in Java?
45. What is difference between an array and a linked list?
46. How do you loop around an array using enhanced for loop?
47. How does Bubble Sort works ?
48. What is the use of Bubble sort?
49. Where Arrays class is present?
50. What is difference between ArrayIndexOutOfBoundsException and ArrayStoreException?