MCA R20 Batch (1st Year 1st Semester)
Java Programming Lab

List of Experiments

1) Write Java Program that uses both recursive and non-recursive functions to print the nth term in the Fibonacci sequence. (Fibonacci Sequence: 0 1 1 2 3 5 8 13 21 34…)

Program:- p1.java

```
import java.util.Scanner;
class p1
{
   public static void main(String args[ ])
   {
      Scanner sc=new Scanner(System.in);
      Fibonacci obj=new Fibonacci ();
      int a,b,c;
      System.out.print("Which term you want to print ");
      a=sc.nextInt();
      b=obj.fibo_recursion(a);
      System.out.println("The "+a+"th Fibonacci term using recursion is: "+b);
      c=obj.fibo_non_recursion(a);
      System.out.println("The "+a+"th Fibonacci term using non recursion is: "+c);
   }
}
class Fibonacci
{
   int a=0;
   int b=1;
   int c;
   int fibo_recursion (int n)
   {
      if(n == 1)
          return 0;
      else if(n == 2)
         return 1;
      else
         return fibo_recursion(n-1) + fibo_recursion(n-2);
   }
   int fibo_non_recursion (int n)
   {
      if(n==1)
        return a;
      else if(n==2)
        return b;
      else
      {
        for(int i=3 ; i<=n ; i++)
        {
          c=a+b;
          a=b;
          b=c;
        }
        return c;
      }
   }
}
```

2) Write Java Program to print all prime numbers up to given integer.
(Prime Numbers: 2 3 5 7 11 13 17 19…)

Program:- p2.java

```
import java.util.Scanner;
class p2
{
  public static void main(String args[ ])
  {
    int i, j, n, f;
    Scanner sc=new Scanner(System.in);
    System.out.print("Enter n value ");
    n=sc.nextInt();
    System.out.println("Prime Numbers upto "+n+" are");
    for(i=2; i<n; i++)
    {
      f=0;
      for(j=2; j<=i/2; j++)
      {
        if(i%j == 0)
        {
           f=1;
           break;
        }
      }
      if(f==0)
        System.out.println(i);
    }
  }
}
```

3) Write Java Program to check whether the given string is palindrome or not.
(Ex: - “abba”, “liril”, “eye”, “madam”, “level”, “radar”, “mom”, “refer”, ….)

Program:- p3.java

```
import java.util.Scanner;
class p1
{
  public static void main(String args[])
  {
    String str, rev = "";
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter a String ");
    str = sc.nextLine();

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

    System.out.println("Reverse String = " + rev);
    if (str.equals(rev))
      System.out.println(str+" is a Palindrome");
    else
      System.out.println(str+" is Not a Palindrome");
   }
}
```

4) Write Java Program to sort given list of names in alphabetical order.
Program:- p4.java

```
import java.util.Scanner;
class Sorting
{
  void sortStrings()
  {
     Scanner sc = new Scanner(System.in);
     System.out.print("How many Names ");
     int i, j, n;
     n = sc.nextInt();
     String[] str = new String[n];
     System.out.println("Enter " + n + " Strings");
     for(i = 0; i < n; i++)
     {
       str[i] = new String(sc.next());
     }
     for(i = 0; i < n; i++)
     {
        for(j = i+1; j < n; j++)
        {
        if(str[i].compareTo(str[j])>0)
        {
            String temp = str[i];
            str[i] = str[j];
            str[j] = temp;
        }
        }
     }
     System.out.println("Given Strings in Alphabetical Order");
     for(i=0; i<n ; i++)
       System.out.println(str[i]);
  }
}
class p4
{
  public static void main(String args[])
  {
     Sorting obj = new Sorting();
     obj.sortStrings();
  }
}
```

5) Write Java Program to illustrates how runtime polymorphism is achieved. (Method Overriding)

Program:-p5.java

```
public class  p1
{
    public static void main(String[] args)
    {
      Vehicle obj = new Bike();
      obj.speed();
    }
}

class Vehicle
{
   public void speed()
   {
      System.out.println("Vehicle speed = 60km/h");
   }
}

class Bike  extends  Vehicle
{
    public void speed()
   {
      System.out.println("Bike speed = 40km/h");
   }
}
```

6) Write a Java Program to create and demonstrate a user defined package.

Program1: - MyMath.java

```
package  mypack;
public class  MyMath
{
   public void  sum(int a, int b)
   {
      int c=a+b;
      System.out.println("Addition = " + c);
   }
}
```

Note:

Compile package program with -d (it represents destination directory) and . (it represents the current folder/Path also can be mentioned in place of “.”).

```
Syntax:-       javac   -d   .   <Filename.Java>
Example: -     javac   -d   .   MyMath.java
```

Program2: - p6.java

```
import  mypack.MyMath;
import  java.util.Scanner;
public class  p6
{
      public static void main(String args[])
      {
            MyMath obj = new MyMath();
            Scanner sc=new Scanner(System.in);
            int n1,n2;
            System.out.println("Enter two numbers\n");
            n1=sc.nextInt();
            n2=sc.nextInt();
            obj.sum(n1,n2);
      }
}
```

Note: -

```
Compile  : javac  p6.java
Run      : java p6
```

7) Write Java Program, using StringTokenizer class, which reads a line of integers and then displays each integer and find sum of all integers.

Program:-p7.java

```
import java.util.Scanner;
import java.util.StringTokenizer;
class p7
{
  public static void main(String args[])
  {
      int n, sum = 0;
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter some integers with one space gap: ");
      String str = sc.nextLine();
      StringTokenizer st = new StringTokenizer(str);
      System.out.println("Given Integers are");
      while (st.hasMoreTokens())
      {
        String temp = st.nextToken();
        n = Integer.parseInt(temp);
        System.out.println(n);
        sum = sum + n;
      }
      System.out.println("Sum of all integers = " + sum);
   }
}
```

8) Write Java Program that reads name of a file from the user and displays the following information about the file.

1. **File exists or Not**
2. **Path of the file**
3. **Whether the file is readable or not**
4. **Whether the file is writable or not**
5. **Total length of the file in bytes**

Program: - p8.java

```
import java.io.File;
import java.util.Scanner;
class P8
{
  public static void main(String args[ ])
  {
      String fn;
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter a file name ");
      fn=sc.next();
      File f1 = new File(fn);
      System.out.println("File Name: " + f1.getName());
      System.out.println("Absolute Path: " + f1.getAbsolutePath());
      System.out.println(f1.exists() ? "The file exists" : "The file does not exist");
      System.out.println(f1.canWrite() ? "It is  writeable" : "It is  not writeable");
      System.out.println(f1.canRead() ? "It is  readable" : "It is  not readable");
      System.out.println(f1.isDirectory() ? "It is a directory " : "It is not a directory");
      System.out.println(f1.isFile() ? "It is  normal file" : "It might be a named pipe");
      System.out.println(f1.isAbsolute() ? "It is  an absolute file" : "It is  not absolute file");
      System.out.println("File size: " + f1.length() + " Bytes");
  }
}
```

9) Write Java Program to display number of characters, words and lines in a text file. Also print the contents of the file.

```
import java.util.Scanner;
import java.io.*;
class P9
{
   public static void main(String args[ ]) throws IOException
   {
     int nl=1,nw=0,nc=0;
     char ch;
     String str;
     Scanner sc = new Scanner(System.in);
     System.out.println("Enter File name ");
     str=sc.nextLine();
     FileInputStream f = new FileInputStream(str);
     int n=f.available();
     System.out.println("Contents of the file: -\n");
     for(int i=0;i<n;i++)
     {
        ch=(char)f.read();
        System.out.print(ch);
        if(ch=='\n')
          nl++;
        else if(ch==' ')
          nw++;
     }
     System.out.println("\nNumber of lines : "+nl);
     System.out.println("\nNumber of words : "+(nl+nw));
     System.out.println("\nNumber of characters : "+n);
   }
}
```

10) Create an Applet that displays the content of a text file.

Part1: -
Create the following text file in the same folder in which you want save applet program.
Aditya.txt: -
Aditya is the best educational institution in AP.
Part2: -
Program: - P10.java

```
import java.applet.*;
import java.awt.*;
import java.io.*;
/* <applet code="P10.class" height="300" width="500"> </applet> */
public class P10 extends Applet
{
   String content = "";
   public void init()
   {
      try
      {
          char ch;
          StringBuffer buff = new StringBuffer("");
          FileInputStream fis = new FileInputStream("Aditya.txt");
          while(fis.available()!=0)
          {
             ch = (char) fis.read();
             buff.append(ch);
          }
          fis.close();
          content = new String(buff);
      }
      catch(FileNotFoundException e)
      {
          content = "Cannot find the specified file...";
      }
      catch(IOException i)
      {
          content = "Cannot read from the file...";
      }
  }
  public void paint(Graphics g)
  {
      Font f = new Font("Courier", Font.PLAIN, 30);
      g.setFont(f);
      g.drawString(content,50,50);
  }
}
```

11) /* Write Java Program to Create a calculator using an applet */

Program: - P11.java

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

/*<applet code="P11" width=300 height=300></applet>*/

public class P11 extends Applet implements ActionListener
{
  String msg=" ";
  int v1, v2, result;
  TextField  t1;
  Button b[] = new Button[10];
  Button add,sub,mul,div,clear,mod,EQ;
  char OP;

  public void init()
  {
    Color k=new Color(120,89,90);
    setBackground(k);
    t1 = new TextField(10);
    GridLayout gl = new GridLayout(4,5);
    setLayout(gl);
    for(int i=0 ; i<10 ; i++)
    {
      b[i]=new Button("" + i);
    }
    add=new Button("+");
    sub=new Button("-");
    mul=new Button("X");
    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("X"))
      {
        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("");
       }
   }
 }
```

12) Write a Java Program for handling mouse events.

Program: - P12.java

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

/* <applet code="P12.class" width=300 height=100> </applet> */

public class P12 extends Applet implements MouseListener, MouseMotionListener
{
  String msg = "";
  int mouseX = 0, mouseY = 0; // coordinates of mouse

  public void init()
  {
    addMouseListener(this);
    addMouseMotionListener(this);
  }

  public void mouseClicked(MouseEvent me)
  {
      mouseX = 0;
      mouseY = 10;
      msg = "Mouse clicked.";
      repaint();
  }

  public void mouseEntered(MouseEvent me)
  {
     mouseX = 0;
     mouseY = 10;
     msg = "Mouse entered.";
     repaint();
  }

  public void mouseExited(MouseEvent me)
  {
     mouseX = 0;
     mouseY = 10;
     msg = "Mouse exited.";
     repaint();
  }

  public void mousePressed(MouseEvent me)
  {
        mouseX = me.getX();
        mouseY = me.getY();
        msg = "Pressed";
        repaint();
    }

    public void mouseReleased(MouseEvent me)
    {
        mouseX = me.getX();
        mouseY = me.getY();
        msg = "Released";
        repaint();
    }

    public void mouseDragged(MouseEvent me)
    {
        mouseX = me.getX();
        mouseY = me.getY();
        msg = "*";
        showStatus("Dragging mouse at " + mouseX + ", " + mouseY);
        repaint();
    }

     public void mouseMoved(MouseEvent me)
     {
         showStatus("Moving mouse at " + me.getX() + ", " + me.getY());
     }

     public void paint(Graphics g)
     {
         g.drawString(msg, mouseX, mouseY);
     }
}
```

13) Write Java Program to demonstrating thread synchronization in multithreading
Program: - p13.java

```
import java.lang.*;
class College
{
  public synchronized void classRoom (String fn)
   {
        for (int i=1 ; i<10 ; i++)
        {
           System.out.println(i + " class taken by " + fn);
           try
           {
              Thread.sleep(1000);
           }
           catch(InterruptedException  e){}
        }
        System.out.println(fn + " task completed\n");
   }
}
class MyThread extends  Thread
{
    College  c;
    String faculty;
    MyThread(College obj, String name)
    {
        c=obj;
        faculty=name;
    }
    public void run()
    {
       c.classRoom(faculty);
    }
}
class SyncDemo
{
   public static void main(String args[])
   {
       College  x = new College();
       MyThread t1 = new MyThread(x,"Babuji Sir");
       MyThread t2 = new MyThread(x,"Kumar Sir");
       MyThread t3 = new MyThread(x,"Pradeep Sir");
       t1.start();
       t2.start();
       t3.start();
   }
}
```

14)Write a Java Program to illustrate user defined Exception Handling (also make use of throw, throws).

```
import java.util.Scanner;
class InvalidBalanceException extends Exception
{
   String msg;
   InvalidBalanceException(String msg)
   {
      this.msg=msg;
   }
   public String toString()
   {
      return msg;
   }
}
class P14
{
   public static void main(String[] args) throws InvalidBalanceException
   {
      Scanner sc=new Scanner(System.in);
      float tf,bf;
      System.out.print("Enter Total Fee:");
      tf=sc.nextFloat();
      System.out.print("Enter Balance Fee:");
      bf=sc.nextFloat();
      try
      {
         if(tf<bf)
         {
            throw new InvalidBalanceException ("InvalidBalanceException ");
         }
         else if(bf==0)
             System.out.println("OK ... We will issue your hall ticket within 10 minutes");
         else
             System.out.println("Within two days you should clear this due amount");
      }
      catch(InvalidBalanceException e)
      {
         System.out.println(e);
      }
   }
}
```

15) Write Java Program to implement Queue, using user defined Exception Handling (also make use of throw and throws)

Program: - p15.java

```
import java.util.Scanner;
import java.lang.Exception;
class  QueueOverFlowException extends Exception
{
}
class  QueueUnderFlowException extends Exception
{
}
class Queue
{
      int n = 5;  // Maximum size of the queue.
      int[] q=new int[n];
      int front = - 1;
      int rear = - 1;
      void enqueue(int  e) throws QueueOverFlowException
     {
        if (rear == n-1)
            throw new QueueOverFlowException ();
        else
        {
            rear++;
            q[rear] = e;
            System.out.println("Element Inserted");
        }
    }
    void dequeue() throws QueueUnderFlowException
    {
        if (front >= rear)
             throw new QueueUnderFlowException();
        else
        {
            front++;
            int  e=q[front];
            System.out.println("Element deleted from queue is : "+ e);
        }
    }

    void display()
    {
        if (front == rear)
        {
            System.out.println("Queue is empty");
            return;
```

```
       }
       else
       {
            System.out.println("Elements qe : ");
            for (int i = front+1; i <= rear; i++)
              System.out.print(q[i]+" ");
            System.out.println();
        }
    }
}
class QDemo
{
    public static  void main(String qgs[])
    {
        Scanner sc=new Scanner(System.in);
        Queue  x = new Queue();
        System.out.println("1: Inserting element to queue(enqueue)");
        System.out.println("2: Deleting element from queue(dequeue)");
        System.out.println("3: Display all the elements of queue");
        System.out.println("4: Exit");
        int ch;
        do
        {
            System.out.println("Enter your choice : ");
            ch=sc.nextInt();
            switch (ch)
            {
                    case 1:
                        System.out.println("enter element to be inserted:");
                        int item=sc.nextInt();
                        try
                       {
                            x.enqueue(item);
                        }
                        catch(QueueOverFlowException e)
                        {
                            System.out.println("Queue is Overflow Not possible to insert new
                                     element into the queue");
                        }
                        break;
                case 2:
                    try
                    {
                       x.dequeue();
                    }
                    catch(QueueUnderFlowException e)
                    {
                        System.out.println("Queue is Underflow Not possible to perform delete
                                                        operation");
                    }
                    break;
                case 3:
                    x.display();
                    break;
                case 4: System.out.println("The End");
                    break;
                default: System.out.println("Invalid choice");
            }
        }
        while(ch!=4);
    }
}
```