Department of Computer Science Engineering

LAB MANUAL

JAVA LAB

GOVT.COLLEGE OF ENGINEERING KALAHANDI

GOVERNMENT COLLEGE OF ENGINEERING KALAHANDI
BHAWANIPATNA

PREPARED BY MR.GOPAL BEHERA, ASST. PROFESSOR, CSE

INDEX

1. STANDARD OPERATING PROCEDURE – SOP

a) Explanation on today's experiment by the concerned faculty using PPT/White Board covering the following aspects: 30 min.

1) Name of the experiment/Aim
2) Software/Hardware required
3) Algorithm/Source code
4) Test Data
   I/P: data set
   O/P: Result

b) Writing of source program by the students 90 min.

c) Compiling and execution of the program 60 min.

Writing of the experiment in the Observation Book:

The students will write the today's experiment in the Observation book as per the following format:

a) Name of the experiment/Aim
b) Software/Hardware required
c) Algorithm/Source Program
d) Test Data/Results for different data sets
   I/P: data set
   O/P:
e) Signature of the Faculty

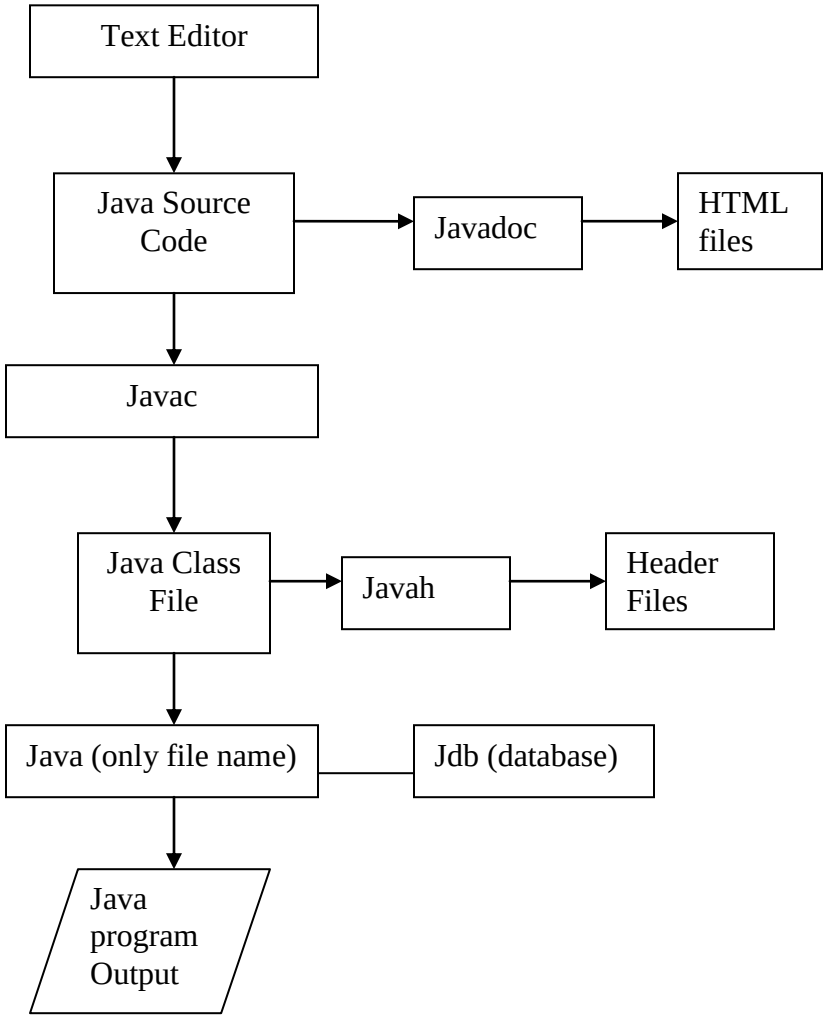
2. LAB OBJECTIVE

1) Introduction to object oriented programming concepts- java as an object oriented programming language. Introduction to java application and applets-control structures-methods-arrays.
2) Object based and objects oriented programming creating packages-using overloaded constructors-static class variables-data abstraction and information hiding-relation between super class objects and subclass objects composition verses inheritance-polymorphism-dynamic method binding abstract super classes and concrete super classes –inheriting interface-use of inner classes and wrapper classes.
3) Role of object oriented programming in designing GUI –Graphs and Java20overview of swing- event handling, adapter classes and layout managers. Advance GUI components-JPopup Menus- JDesktopPane- advance layout managers.
4) Exception handling and multithreading in object oriented programming- When exception handling should be used-java exception handling – exceptions and inheritance-multithreading in java-thread synchronization-daemon threads Runnable interface- Files and streams in java
5) Network and Database handling through object oriented programming –using JOSC – processing queries-overview of servlet –introduction to networking –establishing a simple server and a client – introduction to RMI – implementing the remote interface.

3. List of Lab Exercises

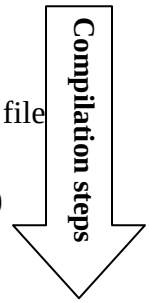
20	Write an applet That computes the payment of a loan based on the amount of the loan the interest rate and the no. of months. It takes one parameter from the browser: monthly rate if true, they interest rate is per month otherwise the interest rate is per annual.	32
Week-9		
21	Write a Program That works as a simple calculator using Grid layout to arrange buttons for the digits and +,-,* % operations. Add a text filed to print the result.	32
22	Write an applet To handling the mouse events	33
Week-10		
23	Write a Program That lets the user to create pie charts ,design your own interface	
24	Write a Program That allow user to draw the line, rectangle and oval	35
Week-11		
25	Write a Program That implements the client/server application. The client sends the data to the server; the client receives the data, uses it to produce a result and then sends the result back to the client. The client displays the result on the console. Ex: The data sent by the client is radius of a circle and the result produced by the server is area of a circle	37

Process of building and running java application programs:



The way these tools are applied to build and run application programs is create a program. We need create a source code file using a text editor. The source code is then compiled using the java compiler javac and executed using the java interpreter java. The java debugger jdb is used to find errors. A complied java program can be converted into a source code.

INTRODUCTION, COMPILING & EXECUTING A JAVA ROGRAM.



Step1: Open notepad.

Step2: Create new file and write your java program.

Step3: Save the file as **classname.jave** for example if your class name is test then your file name should be **test.java**

Step4: Open command prompt (Go to start menu->run->type cmd then enter key press)

Step5: Type **javac filename** and press enter.

If everything is ok the compiler creates a file called fliname.class which contains byte code.

Step6: To run /to see the output type java class name. (This step is the Executing java program)

AIM:

Name of the Experiment No1:

1. Write a java program to find the simple interest.

HW/SW requirements:

Processor	:	AMD Athelon ™ 1.67 GH_z
RAM	:	256 MB
Hard Disk	:	40 GB

JDK is required, with compilers.

Algorithm:

1. Start the program.
2. Declare the variables and assume values
3. Interest To pay=principle*time*rate/100;
4. Print the values.
5. End of class and main method.
6. stop the process

Test Data:

a) Valid data sets: 25000 int value as principal, 12.5f float value as rate, 2.75 as double as interest To Pay.
b) Limiting value sets: Only integer values can be stored in the var. principal.
c) Invalid data sets: If we assign float value in var. principal error will occur.

Results for different data sets:

I/P: Principal amount is Rs.25000 interest=Rs.8593.75

O/P: Total amount to pay to clear the loan = Rs.33593.75

AIM:

Name of the Experiment No-2:

Write a Java program to convert any number to character number format from 0 to 9

Software/Hardware Requirements:

S/W: JDK1.5(JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

Algorithm:

1. Start the program.
2. Read a string with inputstreamReader(System.in).
3. convert the string into Integer.parseInt(stdin.readLine());
4. By using switch case (multi way decision statement) when a match is found, that case is executed.
5. Default it is a break statement exit the switch statement.
6. Stop the program.

Test Data:

Valid Data Set: **4**

Invalid Data Set: **-4**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

Enter any positive single digit number:
4
Four
Enter any positive single digit number:
5
Five

AIM:

Name of the Experiment No3:

2. Write a java program to find sum of digits of the given numbers.

Software/Hardware Requirements:

S/W: JDK1.5(JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

Step1:start

Step2:input n,r,s

Step3:s=0

Step4:d=n%10

Step5:s=s+d

Step6:n=n/10

Step7: is n!=0

Step8: repeat step 4 to step 6 till step 7 is true

Step9:when step 7 falls

Step10:print s

Step11:stop

Test Data:

Valid Data Set: **23415**

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

Enter the number whose digits are to be added: 1234
Sum of the digits: 10

AIM:

Name of the Experiment No-4:

Write a java program to find the given factorial numbers.

Software/Hardware Requirements:

S/W: JDK1.5 (JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

Step1: start

Step2: input n,I,f

Step3: f=i=1

Step4: if(i<=n)

Step5: f=f*i

Step6: i=i+1

Step7: repeat from step5 to step6 till steps true

Step8: print f

tep9: stop

Test Data:

Valid Data Set: **5**

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

Enter the number: 5
Factorial: 120
Enter the number: 3
Factorial: 6

AIM:

Name of the Experiment No-5:

Write a java program to check whether the first number is a multiple of second number.

Software/Hardware Requirements:

S/W: JDK1.5 (JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1. Start the program.
2. Read a string with inputstreamReader(System.in).
3. Convert the string into Integer.parseInt(stdin.readLine());
4. By using switch case (multi way decision statement) when a match is found, that case is executed.
5. Default it is a break statement exit the switch statement.
6. stop the program.

Test Data:

Valid Data Set: **2**

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

```
Enter the First number
2
Enter the second number:
4
The first number is not the multiple of second number
Enter the First number
12
Enter the second number
3
```

The first number is the multiple of second number

AIM:

Name of the Experiment No-6:

Write a java program to check given numbers in a sorting order.

Software/Hardware Requirements:

S/W: JDK 1.5 (JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1. Start the program.
2. Create a class and variables with data types.
3. Read a string with DatainputstreamReader(System.in).
4. convert the string into Integer.parseInt(stdin.readLine());
5. for (int i = (ar.length - 1); i >= 0; i--)
6. {
7. for (int j = 1; j ≤ i; j++)
8. {
9. if (ar[j-1] > ar[j])
10. {
11. int temp = ar[j-1];
12. ar[j-1] = ar[j];
13. ar[j] = temp;
14. } } } }
15. Print the concatenation of arrays.
16. Stop the program.

Test Data:

Valid Data Set: **6 9 3 2 0 1**

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

Enter 10 Elements into Array
2
3
4
1
5
7
8
6
3
1
Required Order is
1
1
2
3
4
5
6
7
8

AIM:

Name of the Experiment No-7:

Write a java program to generate the Armstrong number.

Software/Hardware Requirements:

S/W: JDK 1.5 (JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1. Start the program.
2. Create a class and variables with data types.
3. Declaration of the main class.
4. Read a string with DatainputstreamReader(System.in).

5. convert the string into Integer.parseInt(stdin.readLine());
6. while N >0 repeat step 7 ,8 & 9
7. r=N%10
8. sum=sum+r*r*r formula.
9. N=N/10
10. using if else statement check Sum== given Number or not .
11. Stop the program.

Test Data:

Valid Data Set: **153**

Invalid Data Set: **4**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

Enter any Positive Integer Number:
4
Given Number is not Armstrong Number
Enter any Positive Integer Number:
153
Given Number is Armstrong Number

AIM:

Name of the Experiment No-8:

Write a java program to generate the quadratic equation.

Software/Hardware Requirements:

S/W: JDK1.5 (JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1. Start the program. Import the packages.
2. Create a class and variables with data types.
3. Declaration of the main class.
4. Read a string with inputstreamReader(System.in).
5. convert the string into Integer.parseInt(stdin.readLine());
6. By using if(d==0) “Roots are Equal” .
7. if(d>0) “Roots are Real” otherwise ("Roots are Imaginary");
8. End of class and main method.
9. Stop the program.

Test Data:

Valid Data Set: 1 9 4

Invalid Data Set: 2 3

Limiting Data Sets: Integer Range (-32768 to 32767)

Results for different Data Sets

Enter value of a:

2

Enter value of b:

3

Enter value of c:

4

Roots are Imaginary

Enter value of a:

1

Enter value of b:

9

Enter value of c:

4

Roots are Real

Root1 = -0.46887112585072543

Root2 = -8.531128874149275

AIM:

Name of the Experiment No-9:

Write a java program to generate the Fibonacci series, given number of n values.

Software/Hardware Requirements:

S/W: JDK1.5(JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

Step1: start

Step2: read I,x,f,f1,f2

Step3: f=0,f1=1,f2=1

Step4: do

I++

F1=f2

F2=f

F=f1+f2

While (i<=n)

Step5: print f

Step6: stop

Test Data:

Valid Data Set: **12**

Invalid Data Set: **-12**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

Enter the number u want in Fibonacci series:12
0 1 1 2 3 5 8 13 21
Enter the number u want in Fibonacci series:5
0 1 1 2 3

AIM:

Name of the Experiment No-10:

Write a java program to find the prime numbers up to given number.

Software/Hardware Requirements:

S/W: JDK1.5 (JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

Step1: start

Step2: read n value

Step3: for i=1 i<=n

Step4: repeat a b c d e

a) fact equal to 0

b) for j=1,j<=1 repeat c,d

c)if i percentage j equal to zero

d) fact equal to factorial added with one

e) if factorial equal to2print as prime number

step5: display the prime no till nth num

6: stop

Test Data:

Valid Data Set: **2** **3** **5**

Invalid Data Set: 1 4

Limiting Data Sets: Integer Range (-32768 to 32767)

Results for different Data Sets

Enter Positive value: 10

2

3

5

7

Enter Positive value : 2 6 2 3 5

AIM:

Name of the Experiment No-11:

Write java program display the given string is palindrome or not

Software/Hardware Requirements:

S/W: JDK1.5 (JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1. Start the program. Import the packages.
2. Read a string with inputstreamReader(System.in).
3. reverse the string and store into another string
4. if both the strings are equal

   write palandrom

   else

   write not a palandrom
5. End of class and main method.
6. stop the program.

Test Data:

Valid Data Set: **amma**

Invalid Data Set: **ananth**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

Enter a string
madam
Given string is palindrome

Enter a string
cse
Given string is not palindrome

AIM:

Name of the Experiment No-12:
Write a java program arrange strings into ascending order

Software/Hardware Requirements:

S/W: JDK1.5(JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1. Start the program.
2. Create a class and variables with data types.
3. Read a string with DatainputstreamReader(System.in).
4. convert n value to integer
5. repeate '6' until I <n
6. repeate '7' until j <n-i
7. if a[j]<a[j+1] then

   change value from a[j] to a[j+1]

8. Print the array

9. Stop the program.

```
class sorting
   {
      public static void main(String[] input)
      {
              int k=input.length;
              String temp=new String();
              String names[]=new String[k+1];
              for(int i=0;i<k;i++)
              {
                      names[i]=input[i];
              }
              for(int i=0;i<k;i++)
                      for(int j=i+1;j<k;j++)
                      {
                              if(names[i].compareTo(names[j])<0)
                              {
                                      temp=names[i];
                                      names[i]=names[j];
                                      names[j]=temp;
                              }
                      }
   System.out.println("Sorted order is");
      for(int i=0;i<k;i++)
   {
                      System.out.println(names[i]);
   }
    }
   }
```

Test Data:

Valid Data Set: **deva** **ananth** **amma**

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

Enter the string : **deva** **ananth** **amma**

assending order: **amma** **ananth** **deva**

AIM:

Name of the Experiment No-13:

Write a java program to find the product of the matrices by using two dimensional arrays.

Software/Hardware Requirements:

S/W: JDK1.5(JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

Step1: start

Step2:read I,j,k,a[3][3],b[3][2],c[3][2]

Step3: read a[3][3] & b[3][2]

Step 4:i=0,j=0,k=0

Step5: if i<3 then i++ else goto 1

Step6: if j<3 then j++ else goto 5

Step7: if k<3 then k++ else goto 6

Step8: c[i][j]=c[i][j]+a[k][j]*b[i][k]

Step9: print a[i][j],b[i][j],c[i][j]

Step 10: stop

Test Data:

Valid Data Set: 1 2 3 4 5 6 7 8 9

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

Enter the elements of 1st matrix
1
2
3
4
5
6
7

```
8
9
Enter the elements of the 2nd matrix
1
0
0
0
1
0
0
0
1
The productof the matrices is
 1 2 3
 4 5 6
 7 8 9
```

AIM:

Name of the Experiment No-14:

Write java program display the file is exists and number of bytes in the given file

Software/Hardware Requirements:

S/W: JDK1.5 (JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1. Start the program, import the packages.
2. create an object of 'fi' using fileinputstream class
3. if check the file exit then

   print 'file is exit'

   else

   print 'file does not exit
4. Print the number of bytes of given files is using fi.avail() function
5. Stop the program

Test Data:

Valid Data Set: C:\DOCUME~1\student\Desktop>java p15

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

C:\DOCUME~1\student\Desktop>java p15
number of bytes in this file is:249

AIM:

Name of the Experiment No-16:

Write java program display the no of lines,words and characters of a given file

Software/Hardware Requirements:

S/W: JDK1.5(JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1) Start the program, import the packages.
2) create an object of 'fi' using fileinputstream class
3) compute count=0,word=0,line=1,space=0
4) repeate '4' until file is empty
5) if fi.read() equal to new line then

   line++

   else

   fi.read() equal to space then

   word++

   else

   char++
6) print word,line,char
7) Stop the program

8) Start the program, import the packages.

9) create an object of 'fi' using fileinputstream class

10) compute count=0,word=0,line=1,space=0

11) repeate '4' until file is empty

12) if fi.read() equal to new line then
line++
else
fi.read() equal to space then
word++
else
char++

13) print word,line,char

14) Stop the program

Test Data:

Valid Data Set: C:\DOCUME~1\student\Desktop>java p10

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets
C:\DOCUME~1\student\Desktop>java p16
nof charecters:227 words:23 lines14of the given file

AIM:

Name of the Experiment No-17:

Creating thread by implementing Runnable interface

1. Let your class implement "Runnable" interface.

2. Now override the "public void run()" method and write your logic there (This is the method which will be executed when this thread is started).

That's it, now you can start this thread as given below

1. Create an object of the above class

2. Allocate a thread object for our thread

3. Call the method "start" on the allocated thread object.

```
public class FirstThread implements Runnable
{

  //This method will be executed when this thread is executed
  public void run()
  {

    //Looping from 1 to 10 to display numbers from 1 to 10
    for ( int i=1; i<=10; i++)
    {
        //Displaying the numbers from this thread
        System.out.println( "Messag from First Thread : " +i);

       /*taking a delay of one second before displaying next number
        *
        * "Thread.sleep(1000);" - when this statement is executed,
        * this thread will sleep for 1000 milliseconds (1 second)
        * before executing the next statement.
        *
        * Since we are making this thread to sleep for one second,
        * we need to handle "InterruptedException". Our thread
        * may throw this exception if it is interrupted while it
        * is sleeping.
        *
        */
        try
        {
           Thread.sleep (1000);
        }
        catch (InterruptedException interruptedException)
        {
           /*Interrupted exception will be thrown when a sleeping or waiting
            *thread is interrupted.
            */
            System.out.println( "First Thread is interrupted when it is
sleeping" +interruptedException);
        }
    }
  }
}
public class SecondThread implements Runnable
{

   //This method will be executed when this thread is executed
   public void run()
```

```
   {

      //Looping from 1 to 10 to display numbers from 1 to 10
      for ( int i=1; i<=10; i++)
      {
         System.out.println( "Messag from Second Thread : " +i);

        /*taking a delay of one second before displaying next number
         *
         * "Thread.sleep(1000);" - when this statement is executed,
         * this thread will sleep for 1000 milliseconds (1 second)
         * before executing the next statement.
         *
         * Since we are making this thread to sleep for one second,
         * we need to handle "InterruptedException". Our thread
         * may throw this exception if it is interrupted while it
         * is sleeping.
         */
         try
         {
             Thread.sleep(1000);
         }
         catch (InterruptedException interruptedException)
         {
            /*Interrupted exception will be thrown when a sleeping or waiting
             * thread is interrupted.
             */
             System.out.println( "Second Thread is interrupted when it is
sleeping" +interruptedException);
         }
      }
    }
}

public class ThreadDemo
{
     public static void main(String args[])
     {
        //Creating an object of the first thread
        FirstThread   firstThread = new FirstThread();

        //Creating an object of the Second thread
        SecondThread   secondThread = new SecondThread();

        //Starting the first thread
        Thread thread1 = new Thread(firstThread);
        thread1.start();

        //Starting the second thread
        Thread thread2 = new Thread(secondThread);
        thread2.start();
     }
}
```

Test Data:

Valid Data Set: C:\DOCUME~1\student\Desktop>java p10

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets
C:\DOCUME~1\student\Desktop>java p16

AIM:

Name of the Experiment No-17:
Write a java applet to display simple message
Software/Hardware Requirements:

S/W: JDK1.5(JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1:start
2: take applets
2 :display
3 End applet

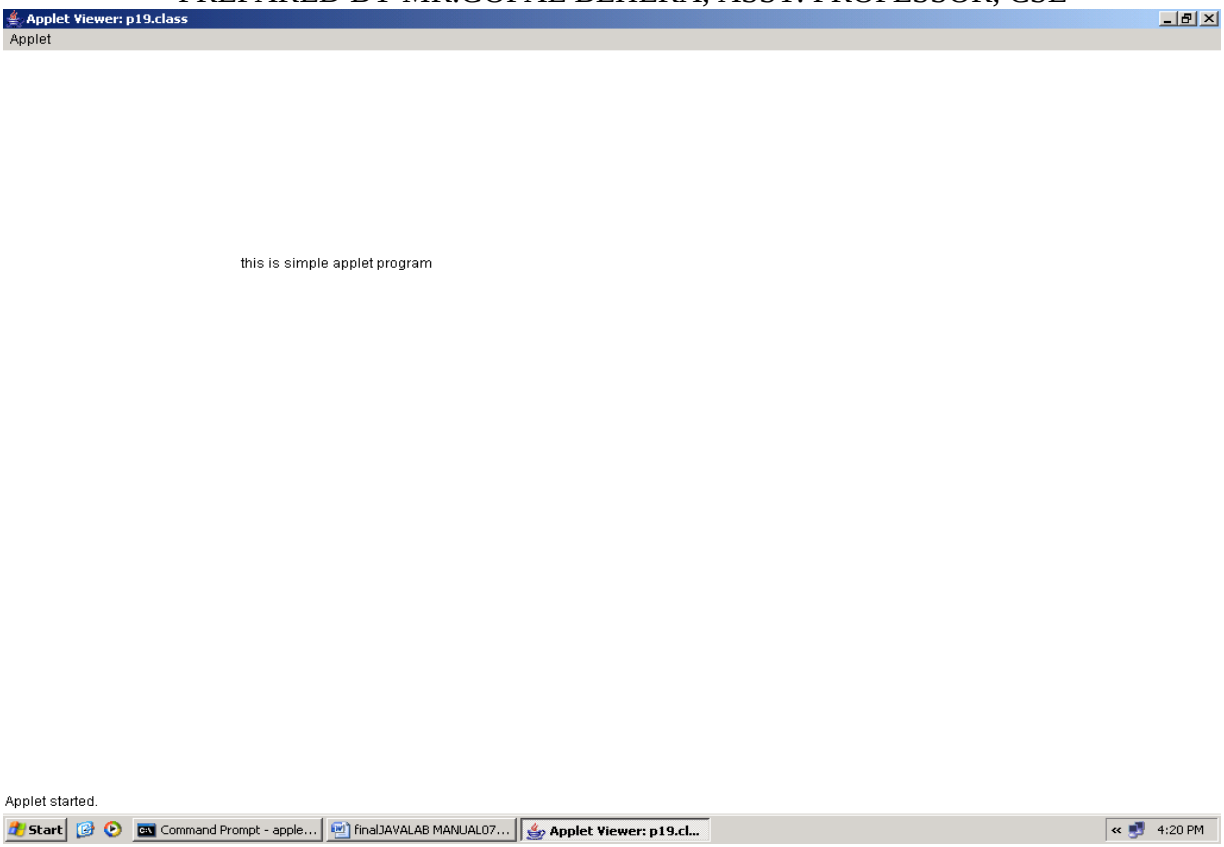
Test Data:

Valid Data Set: **300 400 500**

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets



AIM:

Name of the Experiment No-18:

Write an applet that computes the payment of a loan on the amount of the loan,the interest rate and the number of months.it takes one parameter from the browser:monthly rate,if true,the interest rate is per month otherwise the interest rate is per annual.

Software/Hardware Requirements:

S/W: JDK1.5(JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

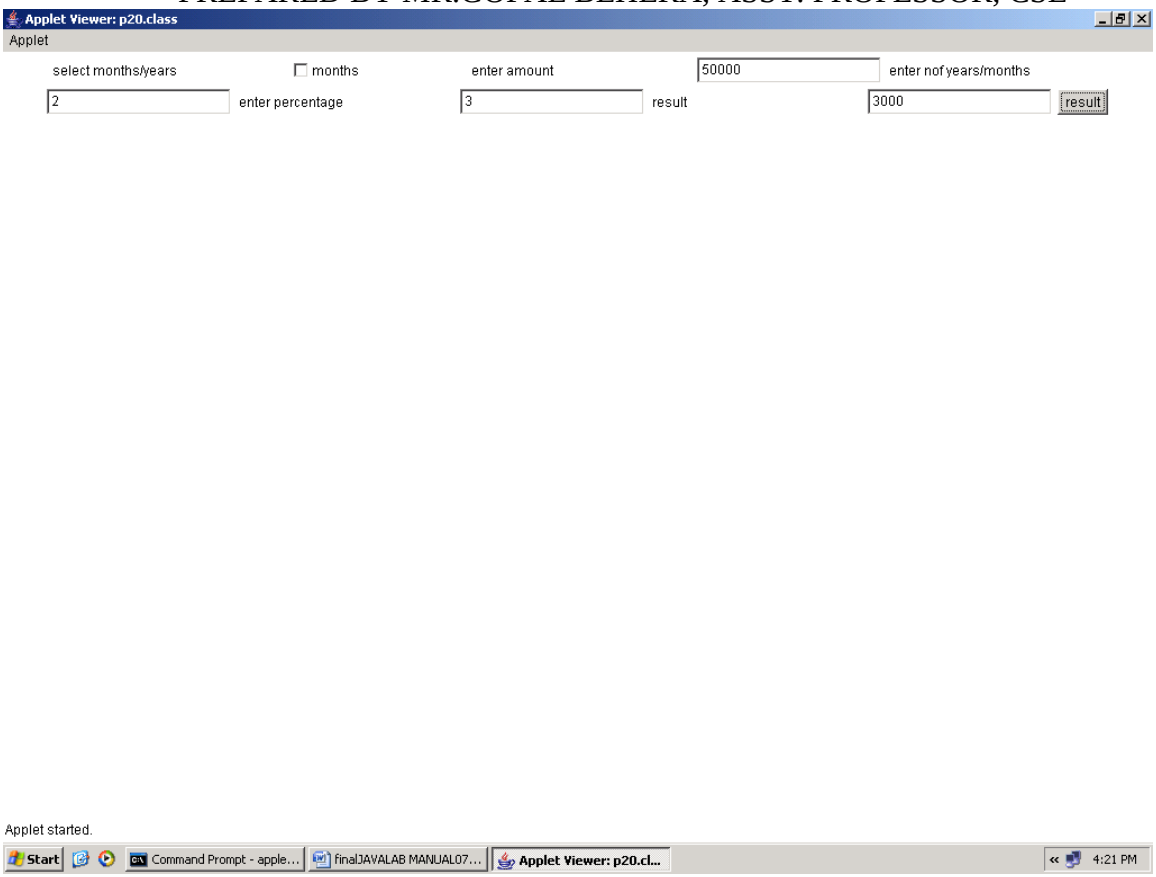
1 start
2 take applet
3 add(new Label("select months/years "));
4 actionPerformed(ActionEvent e)
5 else
6 stop

Integer.parseInt(ta.getText())*Integer.parseInt(tm1.getText())*Integer.parseInt(tr.getText())/1200

Test Data:

Valid Data Set: **24 months**
Invalid Data Set: **Not applicable**
Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets



AIM:

Name of the Experiment No-19:

Write a java program that works as a simple calculator

Software/Hardware Requirements:

S/W: JDK1.5(JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1 start
2 take applet
3 add(new Labels)
4 actionPerformed(ActionEvent e)
5 else
6 prniln i=Integer.parseInt(s2)/Integer.parseInt(s1);
7 stop tf.setText(String.valueOf(i));

Test Data:

Valid Data Set: **press any no: 2 3 4 7**
Invalid Data Set: **Not applicable**
Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

AIM:

Name of the Experiment No20:

Write a java program to implement the APPLET PACKAGES, draw Mouse event handler programs.

Software/Hardware Requirements:

S/W: JDK1.5(JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1. Start the program.
2. Import the packages of applet, awt, awt.event.
3. Create a classes, methods.
4. Mouse moments, mouse Clicked, mouse Pressed, mouse Released, mouse Entered, mouse Exited, mouse Dragged events args.

5. g.drawString() application of Graphical User Interface.
6. while rotating mouse event args.
7. The mouse event arguments execution.
8. Printing in the separated Applet viewer window.
9. Stop the program.

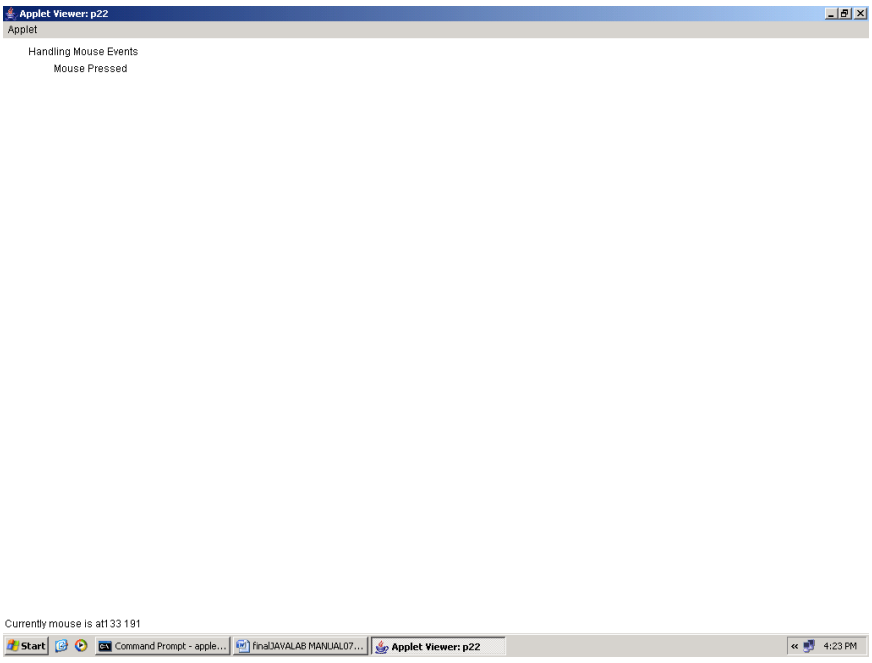
Test Data:

Valid Data Set: **click mouse**

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets



AIM:

Name of the Experiment No21:

Write a java program on interthread communication

Software/Hardware Requirements:

S/W: JDK1.5 (JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1 Start

2 take class

3 for singal wait

4 true 0r false

5 println new InterThread(job)

6 stop

Test Data:

Valid Data Set: **0 1 2**

Invalid Data Set: **Not applicable**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets

put: 0
got: 0
put: 1
got: 1
put: 2
got: 2
put: 3
got: 3
put: 4
got: 4

AIM:

Name of the Experiment No22:

Write a java program to implement the APPLET PACKAGES, draw Lines, Rectangles, Rounded Rectangles, filled Polygons programs.

Software/Hardware Requirements:

S/W: JDK1.5(JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1. Start the program.
2. Import the packages of applet,awt,awt.event.
3. Create a classes: public void paint(Graphics g).
4. Assume the values of string, color and font.
5. g.drawString() application of Graphical User Interface.
6. Printing in the separated Applet viewer window.
7. Stop the program.

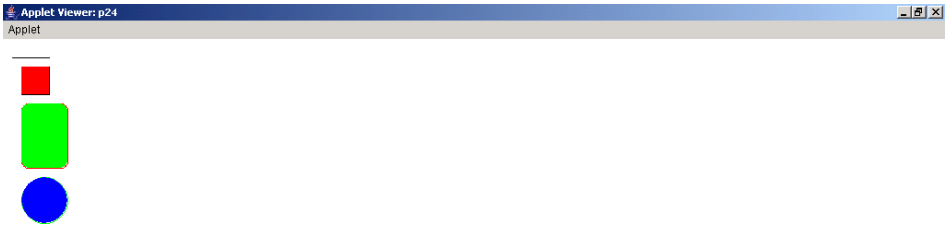
Test Data:

Valid Data Set: **200 400 500**

Invalid Data Set: **1000000000**

Limiting Data Sets: **Integer Range (**-32768 to 32767)

Results for different Data Sets



AIM:

Name of the Experiment No-23:

Write a java program that implements the client/server program

Software/Hardware Requirements:

S/W: JDK1.5 (JAVA), Office XP, Windows NT Server with Service Pack
H/W: Pentium IV, Intel Mother Board Processor, 40 GB HDD, 256 MB RAM

ALGORITHM:

1 Start the program.

2 Import the packages of applet, awt, awt.event.

3 Create a classes, methods.
4 take ServerSocket ssoc=new ServerSocket
5 defalt throws Exception
6 display byte[] data=msg.getBytes();

7 println i=3.14*Float.parseFloat(str)*Float.parseFloat(str);
6 stop program

Test Data:
Valid Data Set: hai
enter socket no:"127.0.0",21

Invalid Data Set: Not applicable

Limiting Data Sets: Integer Range (-32768 to 32767)

Results for different Data Sets
Enter the word : hallow
Given word is: hallow
enter socket no:"127.0.0",21
system no :192.168.10.20 is connected

REFERENCES:

BOOKS:

1. Programming in Java. Second Edition. OXFORD HIGHER EDUCATION. (SACHIN MALHOTRA/SAURAV CHOUDHARY).
2. JAVA Complete Reference (9th Edition) Herbalt Schelidt.
3. Just Java 1.2- Peter van der linder

WEBSITES:

1. **www.java.com**
2. **www.java.sun.com**
3. **www.freewarejava.com**
4. **www.javalobby.org**