Enable | Enlighten | Enrich
(Deemed to be University)
(Under Section 3 of UGC Act 1956)

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed University Established Under Section 3 of UGC Act 1956)
Coimbatore - 641021.
(For the candidates admitted from 2016 onwards)

DEPARTMENT OF COMMERCE (CA)

SUBJECT : PRACTICAL-JAVA
SEMESTER : III
SUBJECT CODE: 16CCP311 **CLASS : II M.COM CA**

QUESTION PAPER

1. Write a program to find the sum of series 1+X+X^2+X^3+…….
2. Write a program to perform string operations.

1. Write a program to find prime or not.
2. Write a program for multiplication tables by multithreading.

1. Write a program to find average of five numbers.
2. Write a program to accept more strings and arrange them in alphabetical order.

1. Define a class for employee with name and data of appointment create employee objects and sort them as per their date of appointment.
2. Write a program to create a window with a background color and display the message.

1. Write a program to find factorial of number using recursion.
2. Write a program to create an exception for mark out of bounds. If mark is greater than 100 throw an exception.

1. Write a program to find simple interest getting values from keyboard.
2. Write a program to accept more strings and arrange them in alphabetical order.

1. Write a program to find maximum of N numbers.
2. Write a program to create an applet and draw the shape.

1. Write a program to find maximum and sum of an array.
2. Write a program to create a window and draw cross lines.

PROGRAM 1:

SUM OF ANY NUMBER OF INTEGERS

AIM:

To write a java program for find the sum of any number of integers

ALGORITHM:

Step 1 : Start the process.

Step 2 : Import input output package.

Step 3 : Create class name as series.

Step 4 : In main function create the object name as “br” buffered reader class.

Step 5 : Enter the value from user using readLine () method.

Step 6 : Get the values for the required variables and convert into integer type.

Step 7 : To calculate mul=mul*x;sum=sum+mul;

Step 8 : Display the result.

Step 9: Stop the Process

SOURCE CODE:

```
import java.util.*;

import java.io.*;

import java.lang.*;

class series

{

public static void main(String[] args) throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the n number of values:");

int n=Integer.parseInt(br.readLine());

System.out.println("Enter the value of x:");

int x=Integer.parseInt(br.readLine());

int i,sum=1,mul=1;

for(i=1;i<=n;i++)

{

mul=mul*x;

sum=sum+mul;

}

System.out.println("Sum of Series=" + sum);

}}
```

OUTPUT:

```
Enter the n number of values: 5

Enter the value of x: 1

Sum of Series=6
```

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 2:

NUMBER IS PRIME OR NOT

AIM:

To check if a number is prime or not by talking the number as input from the keyboard.

ALGORITHM:

```
Step 1 : Start the process.
Step 2 :  Import input output Stream.
Step 3 : Create class name as prime.
Step 4 : In main function create the object name as "br" buffered reader class.
Step 5 : Enter the value from user using readLine () method.
Step 6 : Declare the variables as count=0,1 type int.
Step 7 : To check whether the given number is prime or not using n/2 and n%i==0 using
         for loop.
Step 8 : Stop the process and Display the result.
```

}

SOURCE CODE

```
import java.util.*;

import java.io.*;

import java.lang.*;

class prime

{

public static void main(String [] args) throws IOException

{

int i,n,count=0;

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the value of n :");

n=Integer.parseInt(br.readLine());

for(i=2;i<n;i++)

{

if(n% i==0)

count++;

}

if(count!=0)

{

System.out.println("The given number is not prime number");

}
else

{

System.out.println("The given number is prime number");

}

}

}
```

OUTPUT:

```
Enter the value of n : 1

The given number is prime number
```

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 3:

AVERAGE OF 5 NOS

AIM:

To write a program to find the average of 5 numbers.

ALGORITHM :

```
Step 1  :  Start the process.
Step 2  :  Import input output package.
Step 3  :  Create class name as average.
Step 4 : In main function create the object name as “br” buffered reader class.
Step 5 : Enter the value from user using readLine () method.
Step 6: To calculate sum of values tot=tot+num and average avg=tot/n;
Step 7: Declare the result
Step 8: Stop the Process
```

SOURCE CODE

```
import java.util.*;

import java.io.*;

import java.lang.*;

class avg

{

public static void main(String [] args) throws IOException

{

int num[]=new int[100];

int tot=0,i;

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter,how many number to be calculated:");

int n=Integer.parseInt(br.readLine());

for(i=1;i<=n;i++)

{

num[i]=Integer.parseInt(br.readLine());

}

for(i=1;i<=n;i++)

{

tot=tot+num[i];

}
float avg=tot/n;

System.out.println("The Average of"+ n+" Number is "+ avg);

}

}
```

<u>OUTPUT:</u>

```
Enter,how many number to be calculated: 5

1

2

3

4

5

The Average of 5 Number is 3.0
```

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 4:

EMPLOYEE RECORDS

AIM:

To implement the Employee details using class and sort according to date of Joining.

ALGORITHM:

Step 1: Start the process.

Step 2: Create a class with class name “employee”

Step 3: Define the class and get the details of the employees like Name and date of joining.

Step 4: Define a method to sort those details using swapping according to date of joining

Step 5: Create an object to call those function.

Step 6: Sort the records

Step 7: Display the result.

Step 8: Stop the process.

SOURCE CODE

```
import java.util.*;

import java.io.*;

import java.text.*;

import java.lang.*;

public class Dates

{

public static void main(String args[]) throws IOException,ParseException

{

String name,tname,date;

SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");

BufferedReader v=new BufferedReader(new InputStreamReader(System.in));

Date d, tdate;

int i,j,n;

System.out.println("\n Enter no of employee:");

n=Integer.parseInt(v.readLine());

Employee emp[]=new Employee[n];

System.out.println("\n Enter the "+n+" of Employee Name and Date of join");

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

{

name=v.readLine();
d=new Date(v.readLine());

emp[i]=new Employee(name,d);

}

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

{

for(j=i+1;j<n;j++)

{

if(((emp[i].doj.compareTo(emp[j].doj))>0))

{

tname=emp[i].name;

emp[i].name=emp[j].name;

emp[j].name=tname;

tdate=emp[i].doj;

emp[i].doj=emp[j].doj;

emp[j].doj=tdate;

}

}

}

System.out.println("Employee Name"+"Date of Join");

System.out.println("\n**********"+"\t"+"**********");

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

```
{

System.out.println("\n");

System.out.println(emp[i].name+"\t\t"+sdf.format(emp[i].doj));

}

}

}

class Employee

{

String name;

Date doj=new Date();

public Employee(String n, Date d)

{

name=n;

doj=d;

}

}
```

OUTPUT:

```
Enter no of employee: 2

 Enter the 2 of Employee Name and Date of join

Abi

02/02/2007

Aruna

03/05/2008

Employee NameDate of Join

**********      **********

Abi             02/02/2007

Aruna           05/03/2008
```

<u>RESULT :</u>

The above program has been executed successfully and the output is verified.

PROGRAM 5:

FACTORIAL USING RECURSION

AIM:

To write a program to find the factorial of n number using recursion.

ALGORITHM:

Step 1: Start the process.

Step 2 : Import input output Stream.

Step 3 : Create class name as fact

Step 4 : In main function create the object name as “br” buffered reader class.

Step 5 : Enter the value from user using readLine () method.

Step 6: Factorial of n numbers is found by the formula, n*fact (n-1).

Step 7: Result is displayed.

Step 8: Stop the process.

SOURCE CODE

```
import java.io.*;

class fact

{

public static void main(String args[])throws IOException

{

BufferedReader s=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the n Number:");

int a=Integer.parseInt(s.readLine());

int res=fact(a);

System.out.println("Factorial of given numbers:"+ res);

}

static int fact(int b)

{

if(b<=1)

{

return 1;

}

else

{

return b*fact(b-1);}}}
```

<u>OUTPUT:</u>

```
Enter the n Number: 5

Factorial of given numbers: 120
```

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 6:

SIMPLE INTEREST

AIM:

To write a program to find the simple interest.

ALGORITHM:

```
Step 1 : Start the process.
Step 2 :  Import input output Stream.
Step 3 : Create class name as simple.
Step 4 : In main function create the object name as “br” buffered reader class.
Step 5 : Enter the value from user using readLine () method.
Step 6: Calculate si=pnr/100
Step 7: Declare the results.
Step 8:Stop the Process
```

SOURCE CODE

```
import java.io.*;

class simple

{

public static void main(String args[]) throws IOException

{

int si,p,n,r;

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the p value:");

p=Integer.parseInt(br.readLine());

System.out.println("Enter the n value:");

n=Integer.parseInt(br.readLine());

System.out.println("Enter the r value:");

r=Integer.parseInt(br.readLine());

si=p*n*r/100;

System.out.println("Simple Interest:" + si);

}

}
```

OUTPUT:

```
Enter the p value: 1000

Enter the n value: 2

Enter the r value: 5

Simple Interest: 100
```

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 7:

SUM OF 'N' NUMBERS

AIM :

To write a program to find the sum and maximum of n numbers.

ALGORITHM :

Step 1 : Start the process.

Step 2 : Import input output Stream.

Step 3 : Create class name as simple.

Step 4 : In main function create the object name as "br" buffered reader class.

Step 5 : Enter the value from user using readLine () method.

Step 6: Add the n number of values and store it in sum.

Step 7:Declare the results and stop the process

}

SOURCE CODE

```
import java.io.*;

class max

{

public static void main(String [] args) throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int i,max=0;

System.out.println("Enter the n number value:");

int n=Integer.parseInt(br.readLine());

int largest;

System.out.println("Enter the value:");

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

{

largest=Integer.parseInt(br.readLine());

if(largest>max)

max=largest;

}

System.out.println("Largest number is:" + max);

}

}
```

OUTPUT:

```
Enter the n number value: 5

Enter the value:

100

200

300

50

60

Largest number is: 300
```

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 8:

SUM AND MAXIMUM OF 'N' NUMBERS

AIM :

To write a program to find sum and maximum of n numbers.

ALGORITHM :

Step 1 : Start the process.

Step 2 : Import input output Stream.

Step 3 : Create class name as simple.

Step 4 : In main function create the object name as "br" buffered reader class.

Step 5 : Enter the value from user using readLine () method.

Step 6: Add the n number of values and store it in sum.

Step 7: Maximum of n numbers is found by checking if condition, if(a[i] > max).

Step 8: If it is true, assign max=a[i].

Step 9: Display the output of sum and max.

Step 10: Stop the process.

SOURCE CODE

```
import java.io.*;

class sum

{

public static void main(String [] args) throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int i, max=0,sum=0;

System.out.println("Enter the n number:");

int n=Integer.parseInt(br.readLine());

int largest;

System.out.println("Enter the value:");

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

{

largest=Integer.parseInt(br.readLine());

sum=sum+largest;

if(largest>max)

max=largest;

}

System.out.println("Largest number is:" + max);
System.out.println("The sum of number is:" + sum);}}
```

OUTPUT:

```
Enter the n number: 5

Enter the value:

100

200

300

50

60

Largest number is: 300

The sum of number is: 710
```

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 9:

STRING FUNCTIONS

AIM:

To write a program to perform string operations.

ALGORITHM:

```
Step 1  : Start the process.
Step 2 :  Create in main function create strings.
Step 3 : In main function create String Buffer class.
Step 4 : To calculate length and capacity using String Buffer class.
Step 5 : To calculate capacity and index using String Buffer class.
Step 6 : To calculate set char and append using String Buffer class.
Step 7 : To calculate insert and delete using String Buffer class.
Step 8 : Enter a string like s2.
Step 9 : To calculate length, equals and concatenation using string class.
Step 10 : Stop the process and Display the result.
```

SOURCE CODE

```
import java.io.*;

import java.lang.*;

class string

{

public static void main(String args[]) throws IOException

{

try

{

String st1,st2;

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.print("enter the string1:");

st1=br.readLine();

System.out.print("enter the string 2:");

st2=br.readLine();

System.out.println("the concatination is:" + st1.concat(st2));

System.out.println("the length of first string is:" + st1.length());

System.out.println("the length of second string is:" + st2.length());

System.out.println("second string are equal or not using equal()");

System.out.println(st1.equals(st2));

System.out.println("second string are equal or not using equalIgnoreCase()");
System.out.println(st1.equalsIgnoreCase(st2));

System.out.print("Lower case and Upper case");

System.out.print("the uppercase of first string is:" + st1.toUpperCase());

System.out.println("the lowercase of second string is:" + st2.toLowerCase());

System.out.println("Replace() concept");

String d,e,st;

System.out.println ("enter the value for string 3:");

String st3=br.readLine();

System.out.println("Which letter to be replace");

d=br.readLine();

System.out.println("What is a new character to be placed");

e=br.readLine();

st=st3.replace(d,e);

System.out.println("the replace:" + st);

System.out.println("Substring concept");

System.out.println(st2.substring(1,3));

String ch;

System.out.println("Which character to be search:");

ch=br.readLine();

System.out. println ("Index of"+ st2.indexOf(ch));

System.out.println("LastIndex of"+ st2.lastIndexOf(ch));
```

```
}

catch(StringIndexOutOfBoundsException ae )

{

}

}

}
```

OUTPUT:

```
Enter the string1: hai

Enter the string 2: Hai

The concatenation is: haiHai

The length of first string is: 3

The length of second string is: 3

Second string are equal or not using equal(): false

second string are equal or not using equalIgnoreCase(): true

Lower case and Upper case

The uppercase of first string is: HAI

The lowercase of second string is: hai

Replace () concept

Enter the value for string 3: jeep

Which letter to be replace: j

What is a new character to be placed: k

The replace: keep

Substring concept: ai

Which character to be search: a

Index of 1

LastIndex of 1
```

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 10:

SORTING STRINGS IN ALPHABETICAL ORDER

AIM:

To write a program to accept more strings and arrange them in alphabetical order.

ALGORITHM:

Step 1: Start the process.

Step 2: Specify class,class name and access modifier.

Step 3: Declare the array of string type.

Step 4: Get the values for the required variables in loop.

Step5: Using compareto method in string, check the two string for equality, lessthan or greaterthan.

Step 6: If first string is greaterthan the second string swap them.

Step 7: Continue the above step till the end of the array.

Step 8: Display the result.

Step 9: Stop the process.

SOURCE CODE

```
import java.io.*;

import java.lang.*;

class sort

{

public static void main(String args[]) throws IOException

{

String tname;

BufferedReader br=new BufferedReader( new InputStreamReader(System.in));

int i,j,n;

System.out.println("enter the n value:");

n=Integer.parseInt(br.readLine());

String name[]=new String[n];

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

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

{

name[i]=br.readLine();

}

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

{

for(j=i+1;j<n;j++)
{
if(name[i].compareTo(name[j])>0)
{
tname = name[i];
name[i] = name[j];
name[j] = tname;
}
}
}
System.out.println("after sorting");
for(i=0;i<n;i++)
{
System.out.println(name[i]);
}
}
}
```

OUTPUT:

```
Enter the n value: 3

Enter the names:

Akhila

Neenu

Bindu

After sorting:

Akhila

Bindu

Neenu
```

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 11:

DRAW CROSS LINES

AIM:

To write a program to create applet window and draw a cross lines.

ALGORITHM:

Step 1: Create a class which is inherited from Applet.

Step 2: Create an object ‘g’ for Graphics

Step 3: Define the paint() method

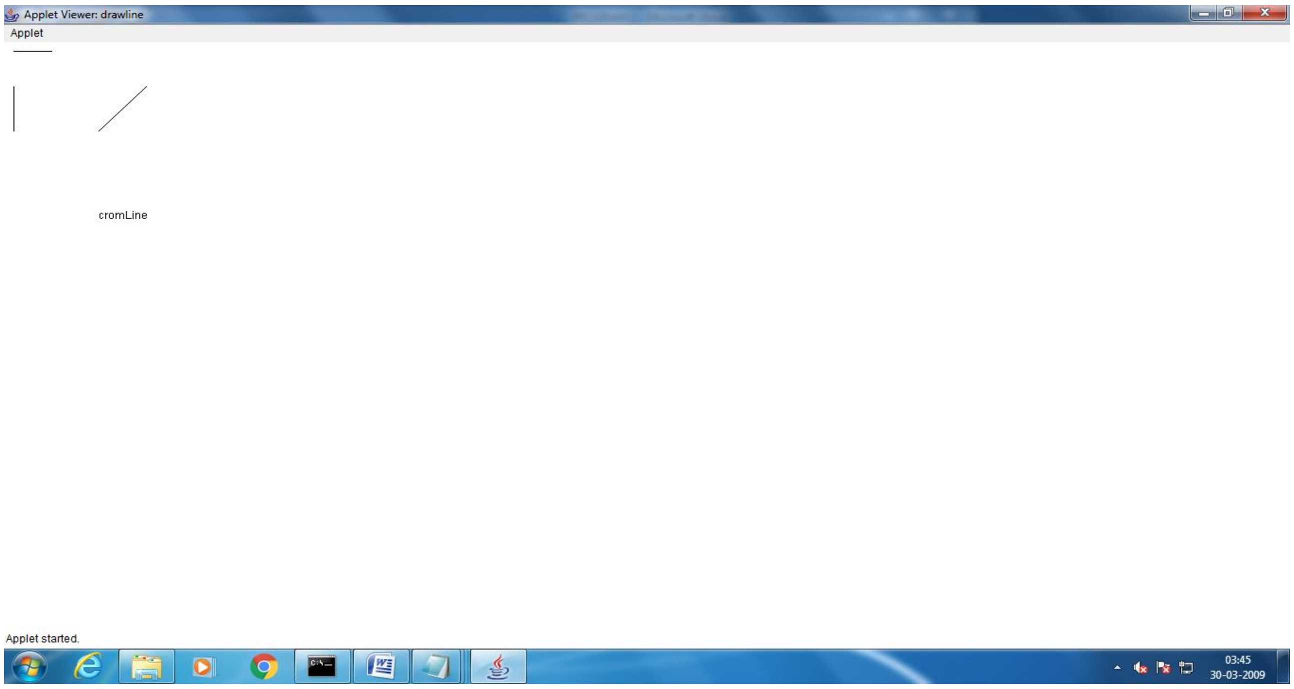
Step 4: To draw cross lines using x,y coordinates value of window.

Step 5: Execute the program using Appletviewer.

SOURCE CODE

```
import java.applet.Applet;

import java.awt.Graphics;

/* <applet code="drawline"width=200 height=200></applet>*/

public class drawline extends Applet

{

public void paint(Graphics g)

{

g.drawLine(150,50,100,100);

g.drawLine(10,50,10,100);

g.drawLine(10,10,50,10);

g.drawString("cromLine",100,200);

}

}
```

OUTPUT:


Applet Viewer: drawline
Applet
cromLine
Applet started.
03:45
30-03-2009

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 12:

SHAPES USING APPLET

AIM:

To write a program to display different shapes in an applet window

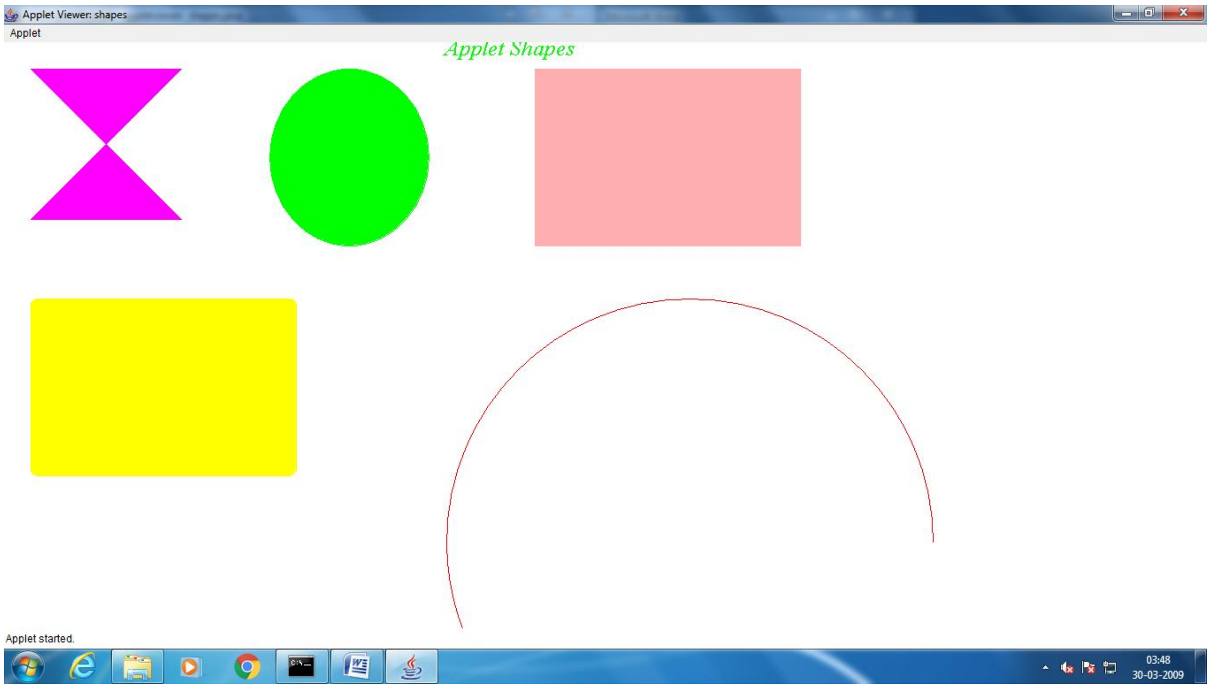
ALGORITHM:

```
Step 1: Start the process.
Step 2: Specify class name, height, width of an applet window in an applet tag.
Step 3: Define class name which extends the Applet class
Step 4: In the paint method using graphics object define parameters for
        different shapes.
Step 5: Display the result.
Step 6: Stop the process.
```

SOURCE CODE

```
import java.awt.*;

import java.applet.*;

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

public class shapes extends Applet

{

public void paint(Graphics g)

{

Font f=new Font("Times New Roman",Font.ITALIC,25);

g.setColor(Color.GREEN);

int x[]={30,200,30,200,70,60};

int y[]={30,30,200,200,70,60};

int num=6;

g.setFont(f);

g.drawString("Applet Shapes", 500,15);

g.setColor(Color.magenta);

g.fillPolygon(x,y,num);

g.drawPolygon(x,y,num);

g.setColor(Color.green);

g.fillOval(300,30,180,200);

g.drawOval(300,30,180,200);
g.setColor(Color.pink);

g.fillRect(600,30,300,200);

g.drawRect(600,30,300,200);

g.setColor(Color.yellow);

g.fillRoundRect(30,290,300,200,15,15);

g.drawRoundRect(30,290,300,200,15,15);

g.setColor(Color.red);

g.drawArc(500,290,550,550,0,250);

}

}
```

OUTPUT:


Applet Viewer: shapes
Applet
Applet Shapes
Applet started.
03:48
30-03-2009

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 13:

DISPLAY MESSAGE USING APPLET

AIM:

To display a message with background color using an Applet.

ALGORITHM:

Step 1: Create a class which is inherited from Applet.

Step 2: Define the init() method to set Foreground and Background Color.

Step 3: Define the start() method to initialize the process.

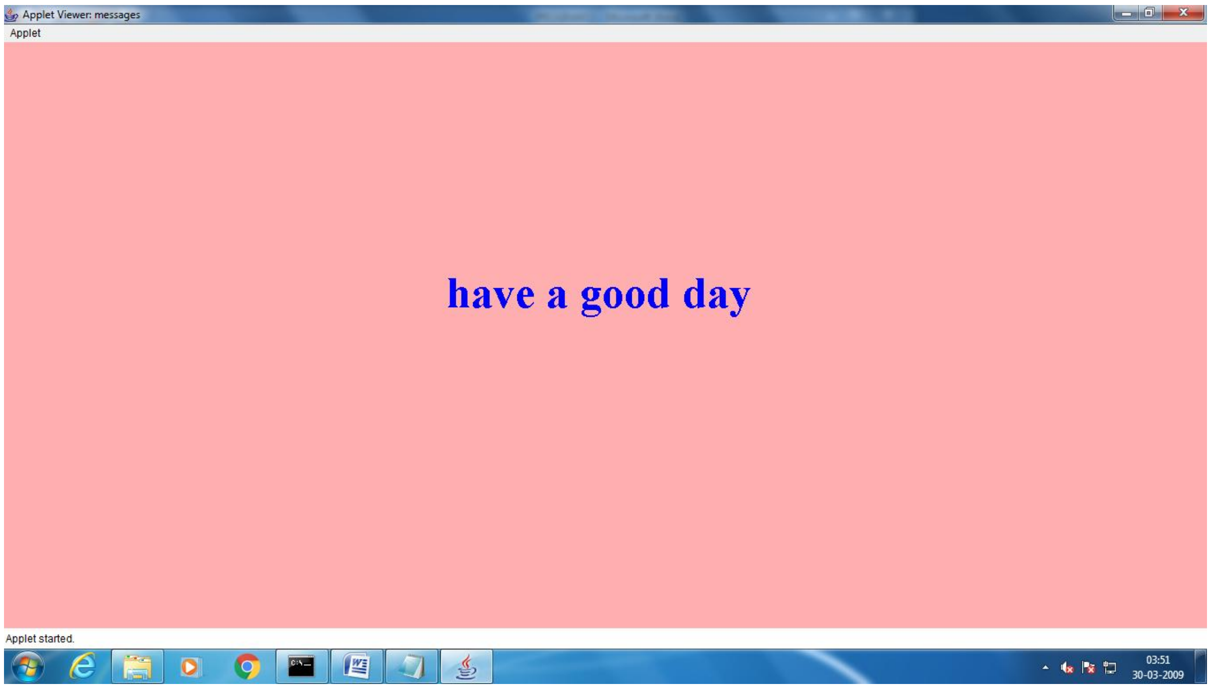
Step 4: Define the paint() method to display the message for certain number of times by passing arguments.

Step 5: Execute the program using Appletviewer.

SOURCE CODE

```
import java.awt.*;

import java.applet.*;

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

public class messages extends Applet

{

public void paint(Graphics g)

{

Font f= new Font("Times New Roman",Font.BOLD,50);

g.setFont(f);

setBackground(Color.pink);

g.setColor(Color.blue);

g.drawString("have a good day",500,300);

}

}
```

OUTPUT:



```
have a good day
```

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 14:

MULTIPLICATION TABLE

AIM:

To write a program to display multiplication table using thread

ALGORITHM:

Step 1: Start the process.

Step 2: Specify class name which implements Runnable interface.

Step 3: Define method name which displays multiplication table.

Step 4: In the main class create the thread for different number and start the thread by using the thread object

Step 5: Display the result.

Step 6: Stop the process.

SOURCE CODE

```
import java.io.*;

class table implements Runnable

{

int n;

public table (int x)

{

n=x;

}

public synchronized void run()

{

for(int i=1;i<=5;i++)

{

System.out.println(i+"*"+n+"="+(i*n));

}

}

}

public class multi

{

public static void main(String args[])

{
```

```
table t1=new table(3);

table t2=new table(4);

table t3=new table(9);

Thread t=new Thread(t1);

t.start();

t=new Thread(t2);

t.start();

t=new Thread(t3);

t.start();

}

}
```

OUTPUT:

```
1*3=3

1*4=4

1*9=9

2*3=6

2*4=8

2*9=18

3*3=9

3*4=12

3*9=27

4*3=12

4*4=16

4*9=36

5*3=15

5*4=20

5*9=45
```

RESULT :

The above program has been executed successfully and the output is verified.

PROGRAM 15:

EXCEPTION HANDLING

AIM:

To write a program to display message when mark exceeds hundred

ALGORITHM:

Step 1: Start the process.

Step 2: Specify class name which implements Exception class.

Step 3: In the main class create try block to check the condition whether mark exceeds 100.

Step 4: Create catch block to catch the exception thrown by the try block

Step 5: Display the result.

Step 6: Stop the process.

SOURCE CODE

```
import java.io.*;

import java.lang.Exception;

class myExcept extends Exception

{

myExcept(String message)

{

super(message);

}

}

class numexp

{

public static void main(String args[]) throws IOException

{

BufferedReader h=new BufferedReader(new InputStreamReader(System.in));

int m1,m2,m3,no;

String name;

System.out.println("enter the register number");

no=Integer.parseInt(h.readLine());

System.out.println("enter the name");
name=h.readLine();

System.out.println("enter the three marks:");

m1=Integer.parseInt(h.readLine());

m2=Integer.parseInt(h.readLine());

m3=Integer.parseInt(h.readLine());

try

{

if(m1>100|m2>100|m3>100)

{

throw new myExcept("marks should be less than 100");

}

}

catch(myExcept e)

{

System.out.println("\n");

System.out.println(e.getMessage());

System.exit(0);

}

System.out.println("name:"+name+"\t number:"+no+"\t mark 1:"+m1+"\t mark 2:"+m2+"\t mark
3:"+m3);

}}
```

OUTPUT:

```
Enter the register number: 101

Enter the name: Kichu

Enter the three marks:

60

70

80

Name: kichu      number: 101     mark 1: 60      mark 2: 70      mark 3: 80

Enter the register number: 102

Enter the name: Kavi

Enter the three marks:

100

200

300

Marks should be less than 100
```

RESULT :

The above program has been executed successfully and the output is verified.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed University Established Under Section 3 of UGC Act 1956)
Coimbatore - 641021.
(For the candidates admitted from 2016 onwards)
DEPARTMENT OF COMMERCE (CA)

16CCP311 **PRACTICAL 5 -JAVA**

Semester – III

L	T	P	C
-	-	4	2

Course Objective

To enable the students to tackle complex programming problems, making good use of the object oriented programming paradigm to simplify the design and implementation process.

Learning Outcomes

- ➢ To gain the practical knowledge on Java.
- ➢ To design and implement the Java programs for Recursion, Array, String Operations and Multithreading.
- ➢ To develop practical skills by using JAVA Program

1. Write a program to find the sum of series 1+X+X^2+X^3+…….
2. Write a program to find prime or not.
3. Write a program to find average of five numbers.
4. Define a class for employee with name and data of appointment create employee objects and sort them as per their date of appointment.
5. Write a program to find factorial of number using recursion.
6. Write a program to find simple interest getting values from keyboard.
7. Write a program to find maximum of N numbers.
8. Write a program to find maximum and sum of an array.
9. Write a program to perform string operations.
10. Write a program to accept more strings and arrange them in alphabetical order.
11. Write a program to create a window and draw cross lines.
12. Write a program to create an applet and draw the shape.
13. Write a program to create a window with a background color and display the message.
14. Write a program for multiplication tables by multithreading.
15. Write a program to create an exception for mark out of bounds. If mark is greater than 100 throw an exception.