CONCORD ARTS AND SCIENCE COLLEGE

CONCORD EDUCITY, MUTTANNUR, (PO) PATTANNUR

FOURTH SEMESTER BACHELOR OF COMPUTER APPLICATION

PRACTICAL RECORD

2020-2021

JAVA PROGRAMMING

CONCORD ARTS AND SCIENCE COLLEGE

CONCORD EDUCITY, MUTTANNUR, (PO) PATTANNUR

CERTIFICATE

It is certified that this is a bonafide record of the original work done by Mr./Mrs.……. Reg.no.....…… of IVth semester BCA in the java Programming and linux adminstration lab during the year 2020-2021.

HOD: **Lecturer in charge:**

Submitted for practical examination held on ……

External Examiner

1.

2.

INDEX

JAVA PROGRAMMING

INDEX
SHELL SCRIPTING

INDEX

LINUX ADMINISTRATION

JAVA PROGRAMMING

1.

AIM

Write a java program to perform various string operations using java class.

PROGRAM

```
import java.i1o.*;

class mystring

{

        void uppercase(String str)

        {

                str=str.toUpperCase();

                System.out.println("The given string in uppercase is "+str);

        }

         void lowercase(String str)

                {

                  str=str.toLowerCase();
                 System.out.println("The given string in lower case
is:\t"+str);
          }

        void cut(String str)

        {

                str=str.trim( );
```

```
            System.out.println("The string after trim is :\t"+str);
        }
        void size(String str)
        {
            System.out.println("The length of the string is:\t
"+str.length());
        }
        void reverse(String str)
        {
            StringBuffer s=new StringBuffer(str);
            s.reverse( );
            System.out.println("The reverse of the string is :\t"+s);
        }
        void search(String str,String ch)
        {
            int c=str.indexOf(ch);
            c=c+1;
            if(c==0)
            {
                System.out.println("Character is not present");
            }
            else
            {
```

```
                System.out.println("The first occurence is:\t "+c);
            }
        }
        public static void main(String args[])
        {
            mystring ms=new mystring( );
            try
            {
                InputStreamReader is=new
InputStreamReader(System.in);
                BufferedReader br=new BufferedReader(is);
                System.out.println("Enter the string");
                String str=br.readLine( );
                ms.uppercase(str);
                ms.lowercase(str);
                ms.size(str);
                ms.cut(str);
                ms.reverse(str);
                System.out.println("Enter the character to
searched");
                String ch=br.readLine( );
                ms.search(str,ch);
            }
```

```
            catch (Exception e)
            {
                System.out.println("There is an error ");
            }
        }
}
```

OUTPUT

```
Enter the string
java
String is :  java
The given string in uppercase is :  JAVA
The length of the string is :  4
The string after trim is :  java
The reverse of the string is : avaj
Enter the character to searched  :  j
 The first occurence is 1
```

2.

AIM

Write java program to implement interface.

PROGRAM

```
import java.io.*;
class Student
{
        int rollno;
         String name;
        void getNumber(int n)
        {
                        rollno=n;
        }
          void  getName(String  str)
           {
                           name = str ;
           }
        void putNumber( )
        {
                        System.out.println("Roll no : "+rollno);
```

```
        }

      void putName( )
        {
                    System.out.println("Name:"+name);
        }
 }
class Test extends Student
{
        float mark1,mark2;
        void getMark(float m1,float m2)
        {
                    mark1=m1;
                    mark2=m2;
         }
        void putMark( )
        {
                    System.out.println("Marks Obtained");
                    System.out.println("Mark1 : "+mark1);
                    System.out.println("Mark2 : "+mark2);
        }
}
interface Sports
{
```

```
        float sportwt=10;
        void putWt( );
}
class Result extends Test implements Sports
{
        float total;
        public void putWt( )
        {
                System.out.println("Sports weight :"+sportwt);
        }
        void display( )
        {
                total=mark1+mark2+sportwt;
                putNumber( );
                   putName( );
                putMark( );
                putWt( );
                System.out.println("Total Score : "+total);
        }
}
class Hybrid
{
```

```
        public static void main(String args[])throws IOException
    {
            Result rs=new Result( );
            InputStreamReader is=new
InputStreamReader(System.in);
            BufferedReader br=new BufferedReader(is);
            int  rno;
            String  str;
            float  m1,m2;
            System.out.println("Enter roll number");
            rno=Integer.parseInt(br.readLine( ));
            System.out.println("Enter Name: ");
            str=br.readLine( );
            System.out.println("Enter 2 marks");
            m1=Float.parseFloat(br.readLine( ));
            m2=Float.parseFloat(br.readLine( ));
            rs.getNumber(rno);
               rs.getName(str);
            rs.getMark(m1,m2);
            rs.display( );
    }
}
```

OUTPUT

Enter roll number

100

Enter Name

Anu

Enter 2 marks

80

90

Roll no : 100

Name : Anu

Marks Obtained

Mark1 : 80.0

Mark2 : 90.0

Sports weight : 10.0

Total Score : 180.0

3.

AIM

Write java program that handles various exceptions.Use try-catch statements.

PROGRAM

```
import java.io.*;
class Execp
{
	public static void main(String args[])
	{
		 int a[ ]={10,5,3};
		try
		{
			int result=a[1]/a[3];
			System.out.println("The result is"+result);
		}
		catch(ArrayIndexOutOfBoundsException AE)
		{
			System.out.println("Array Index is not available"+AE);
		}
		try
		{
```

```
                String str="Concord";
                int l=str.length( );
                System.out.println("Length is"+l);
                char c=str.charAt(10);
                System.out.println("Character is"+c);
            }
            catch(StringIndexOutOfBoundsException SE)
            {
                System.out.println("Reffered Index is not
present"+SE);
            }
        try
            {
                int x=a[0]/(a[1]-5);
            }
        catch(ArithmeticException aa)
            {
                System.out.println("Division by zero error"+aa);
            }
            finally
            {
                System.out.println("I am  always here");

            }
    }
```

}

OUTPUT

Array Index is not availablejava.lang.ArrayIndexOutOfBoundsException: 3

Length is 7

Reffered Index is not presentjava.lang.StringIndexOutOfBoundsException: String index out of range: 10

Division By Zerojava.lang.ArithmeticException: / by zero

I am always here.

4.

AIM

Write java program to implement file I/O operation using java iostreams.

```
 import java.io.*;
class Fileop
{
	public static void main(String args[]) throws IOException
	{
		int ch=0;
		String fname;
		InputStreamReader is=new
InputStreamReader(System.in);
		BufferedReader br=new BufferedReader(is);
		do
		{
			System.out.println("\nMenu \n1.Write to File
			\n2.Read  from File \n3.Exit \n");

			System.out.println("Enter your choice");
			ch=Integer.parseInt(br.readLine( ));
			switch(ch)
			{
```

```
                    case 1: flWrite( );
                            break;
                    case 2: flRead( );
                            break;
                    case 3: System.exit(0);
                    default: System.out.println("Invalid
Entry");
                }
            }
            while(ch<=3);
        }
    public static void flWrite( ) throws IOException
    {
            int c;
            String fname;
            InputStreamReader is=new
InputStreamReader(System.in);
            BufferedReader br=new BufferedReader(is);


            System.out.println("Enter the filename");
            fname=br.readLine();
```

```
                FileOutputStream fw=new
FileOutputStream(fname);
                System.out.println("Enter the content[Press ~ to
stop]\n");
                while((c=br.read( ))!='~')
                {
                        fw.write((char)c);
                }
        }
        public static void flRead( ) throws IOException
        {
                int c;
                String fname;
                InputStreamReader is=new
InputStreamReader(System.in);
                BufferedReader br=new BufferedReader(is);
                System.out.println("Enter the filename");
                fname=br.readLine( );
                File f=new File(fname);



                System.out.println("Content is \t");
                if(!f.exists( ))
```

```
        {
                System.out.println("File not found");
                System.exit(0);
            }
            FileInputStream fr=new FileInputStream(fname);
            while((c=fr.read( ))!=-1)
            {
                System.out.print((char)c);
            }
        }
}
```

OUTPUT

Menu

1.Write to File

2.Read from File

3.Exit

Enter your choice

1

Enter the filename

Java

Enter the content[Press ~ to stop]

Exception

Inheritance

Interface

~

Menu

1.Write to File

2.Read from File

3.Exit

Enter your choice

2

Enter the filename

Java

Content is

Exception

Inheritance

Interface

Menu

1.Write to File

2.Read from File

3.Exit

Enter your choice 3

5.

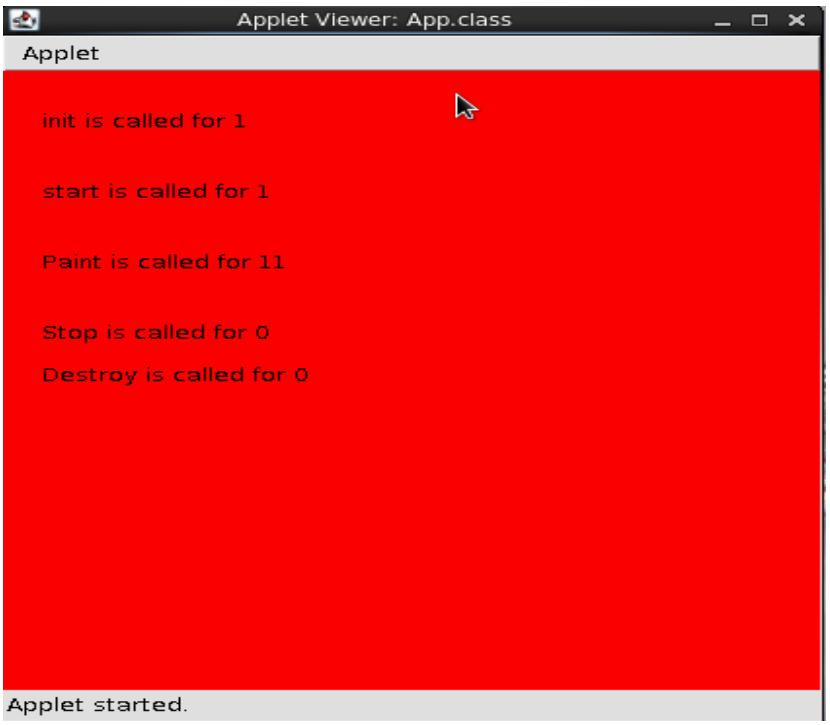
AIM

Write Java program to implement Applet Life cycle.

PROGRAM

```
import java.applet.*;
import java.awt.*;
public class App extends Applet
{
    static int initcall,paintcall,startcall,stopcall,destroycall;
    String name;
    public void init( )
    {
        Color c1=new Color(250,0,0);
        Color c2=new Color(10,40,20);
        setBackground(c1);
        setForeground(c2);
        name="Concord";
        initcall=initcall+1;
    }
    public void start( )
```

```
    {
        startcall=startcall+1;
    }
    public void paint(Graphics g)
    {
        paintcall=paintcall+1;
        g.drawString("init is called for "+initcall,20,40);
        g.drawString("start is called for "+startcall,20,90);
        g.drawString("Paint is called for "+paintcall,20,140);
       g.drawString("Stop is called for" +stopcall,20,190);
        g.drawString("Destroy is called for "+destroycall,20,220);
    }
    public void stop( )
    {
        stopcall=stopcall+1;
    }
    public void destroy( )
    {
        destroycall=destroycall+1;
    }
}
/*<APPLET code="App.class" height=400 width=400> </APPLET>
*/
```


Applet Viewer: App.class
Applet
init is called for 1
start is called for 1
Paint is called for 11
Stop is called for 0
Destroy is called for 0
Applet started.

6.

AIM

Write java program to implement a calculator using suitable AWT controls.

PROGRAM

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class calc extends Applet implements ActionListener
{
    Button n[ ]=new Button[25];
    TextField t;
    double nn,res;
    String op;
    Color c=new Color(100,20,20);
    Color c1=new Color(255,255,255);

    public void init( )
    {
        setBackground(c);
```

```
setLayout(null);
t=new TextField(20);
t.setBounds(50,20,212,27);
add(t);
t.setEditable(false);
for(int i=0;i<=9;i++)
{
        n[i]=new Button(String.valueOf(i));
}
n[10]=new Button(".");
n[11]=new Button("+");
n[12]=new Button("-");
n[13]=new Button("/");
n[14]=new Button("*");
n[15]=new Button("sqrt");
n[16]=new Button("=");
n[17]=new Button("C");
n[18]=new Button("ON");

n[19]=new Button("OFF");
int k=1;
int x=50;
```

```
int y=50;
for(int i=0;i<=19;i++)
{
        if(k%4==0)
        {
                n[i].setBounds(x,y,30,30);
                x=50;
                y=y+40;
        }
        else
        {
                n[i].setBounds(x,y,30,30);
                x=x+60;
        }
        k++;
}
for(int i=0;i<=19;i++)
{

        add(n[i]);
}
for(int i=0;i<=19;i++)
```

```
        {
            n[i].addActionListener(this);
            n[i].setBackground(c1);
        }
}
public void actionPerformed(ActionEvent e)
{
        if(e.getActionCommand().equals("ON"))
        {
            t.setEditable(true);
            t.setText("");
        }
        if(e.getActionCommand( ).equals("OFF"))
        {
            t.setEditable(false);
            t.setText("");
        }
        if(t.isEditable( ))

        {
            if(e.getActionCommand( )=="+")
            {
```

```
        nn=Double.parseDouble(t.getText());
        t.setText("");
        op="plus";
}
else if(e.getActionCommand( )=="-")
{
        nn=Double.parseDouble(t.getText());
        t.setText(" ");
        op="minus";
}
else if(e.getActionCommand( )=="*")
{
        nn=Double.parseDouble(t.getText());
        t.setText("");
        op="mult";
}
else if(e.getActionCommand( )=="/")
{

        nn=Double.parseDouble(t.getText());
        t.setText("");
        op="div";
```

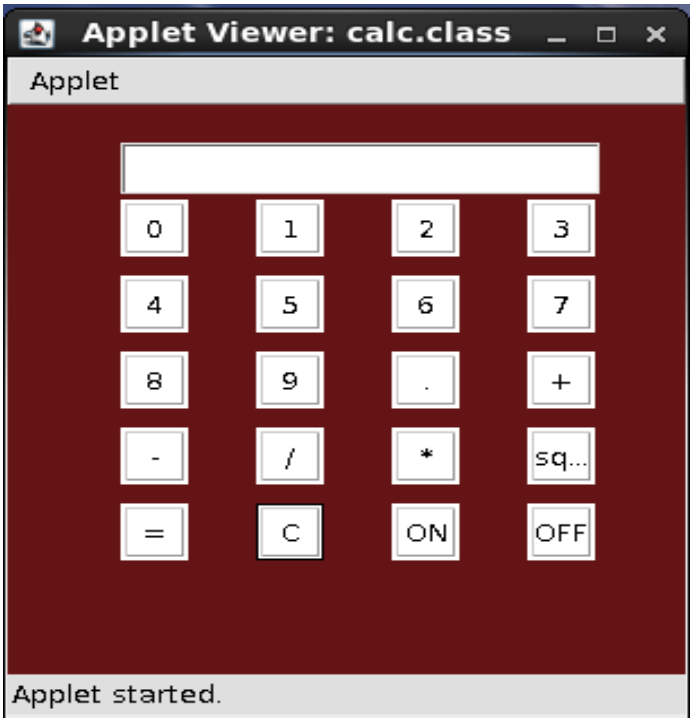
```
}
else if(e.getActionCommand( )=="sqrt")
{
        nn=Double.parseDouble(t.getText());
        t.setText("");
        op="sqrt";
}
else if(e.getActionCommand( )=="=")
{
        if(op.equals("plus"))
        {
                res=nn+Double.parseDouble(t.getText( ));
                t.setText(String.valueOf(res));
        }
        if(op.equals("minus"))
        {
                res=nn-Double.parseDouble(t.getText( ));
                t.setText(String.valueOf(res));

        }
        if(op.equals("div"))
        {
```

```
                    res=nn/Double.parseDouble(t.getText( ));
                    t.setText(String.valueOf(res));
                }
                if(op.equals("mult"))
                {
                    res=nn/Double.parseDouble(t.getText( ));
                    t.setText(String.valueOf(res));
                }
                if(op.equals("sqrt"))
                {
                    res=Math.sqrt(Integer.parseInt(t.getText(
)));
                    t.setText(String.valueOf(res));
                }
            }
            else
            {
                if(e.getActionCommand( )=="C")
                {
                    t.setText("");
                }
                else if((e.getActionCommand( )!="ON")  &&
                        (e.getActionCommand( ) !="OFF"))
```

```
                    {
                        t.setText(t.getText(
)+e.getActionCommand( ));
                    }
                }
            }
        }
}
```

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

OUTPUT



7.

AIM

Write a program to implement packages in java.

```
package mypackage;
public class Calculator
{
   public int add(int a, int b)
   {
      return a+b;
   }

   public int subtract(int a, int b)
   {
      return a-b;
   }
}
```

```
package newpackage;
import mypackage.Calculator;
import java.util.Scanner;
public class Calculate
{
   public static void main(String[] args)
   {
         Calculator calculator = new Calculator();
         System.out.println("Enter two numbers");
         Scanner scanner = new Scanner(System.in);
          int a = scanner.nextInt();
           int b = scanner.nextInt();
```

```
        int added = calculator.add(a,b);
        System.out.println("Sum is "+added);
        int subtracted = calculator.subtract(a,b);
        System.out.println("Subtracted value is "+subtracted);

    }
}
```

OUTPUT

```
Enter two numbers
2
4
Sum is 6
Subtracted value is -2
```

8.

AIM

Program to demonstrate Menu and PopupMenu

PROGRAM

```
import java.awt.*;
import java.awt.event.*;
public class hello extends Frame implements ActionListener
{
   PopupMenu popup;
   MenuBar mb;
   Menu color;
   MenuItem line,circle,square,rectangle,exit,red,green,blue;
   String arg;
   public hello( )
   {
       setTitle("POPUP");
       popup=new PopupMenu("Draw");
       mb=new MenuBar();

       setMenuBar(mb);
       color=new Menu("color");
```

```
line=new MenuItem("line");
circle=new MenuItem("circle");
square=new MenuItem("square");
rectangle=new MenuItem("rectangle");
exit=new MenuItem("exit");
red=new MenuItem("red");
green=new MenuItem("green");
blue=new MenuItem("blue");
popup.add(line);
popup.add(circle);
popup.add(square);
popup.add(rectangle);
popup.add(exit);
add(popup);
color.add(red);
color.add(green);
color.add(blue);
mb.add(color);
line.addActionListener(this);

circle.addActionListener(this);
square.addActionListener(this);
```

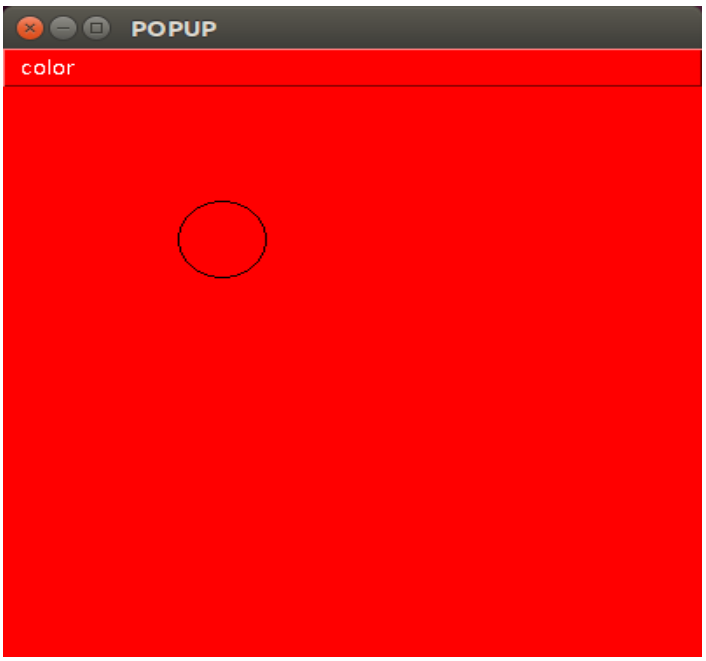
```
        rectangle.addActionListener(this);
        exit.addActionListener(this);
        red.addActionListener(this);
        green.addActionListener(this);
        blue.addActionListener(this);
        addWindowListener(new WH());
        this.addMouseListener(new MouseAdapter()
            {
              public void mouseClicked(MouseEvent me)
                {

popup.show(me.getComponent(),me.getX(),me.getY());
                }
            });
       }
    public void actionPerformed(ActionEvent ae)
    {
        arg=ae.getActionCommand();
        repaint();
    }

    public void paint(Graphics g)
    {
```

```
if(arg.equals("line"))
{
  g.drawLine(100,100,200,100);
}
else if(arg.equals("circle"))
{
  g.drawOval(100,100,50,50);
}
else if(arg.equals("square"))
{
  g.drawRect(100,100,200,200);
}
else if(arg.equals("rectangle"))
{
  g.drawRect(100,100,200,50);
}
else if(arg.equals("red"))
{
  Color c1=new Color(255,0,0);

  setBackground(c1);
}
```

```
    else if(arg.equals("green"))
    {
      Color c1=new Color(0,200,0);
      setBackground(c1);
    }
    else if(arg.equals("blue"))
    {
       Color c1=new Color(0,0,200);
       setBackground(c1);
    }
   else
    {
      System.exit(0);
    }
}
public static void main(String args[])
{
     hello obj=new hello();
     obj.setBounds(1,1,400,400);

     obj.show();
}
```

```
}
class WH extends WindowAdapter
{
    public void windowClosing(WindowEvent we)
    {
        System.exit(0);
     }
}
```

OUTPUT



9.

AIM

Write a java program to demonstrate threads.

PROGRAM

```
import java.io.*;
class A extends Thread
{
	public void run( )
	{
		for(int i=1;i<=5;i++)
		{
			if(i==1)
			{
				yield( );
			}
			System.out.println("From Thread A: i= "+i);
		}
		System.out.println("Exit from A");
	}
}
class B extends Thread
```

```
{
        public void run( )
        {
                for(int j=1;j<=5;j++)
                {
                        System.out.println("Exit Thread B: j= "+j);
                        if(j==3)
                        {
                                stop( );
                        }
                        System.out.println("Exit from B");
                }
        }
}
class C extends Thread
{
        public void run( )
        {
                for(int k=1;k<=5;k++)

                {
                        System.out.println("From thread C: k= "+k);
```

```
                if(k==1)
                {

                        try
                        {
                                sleep(1000);
                        }
                        catch(Exception e)
                        {
                                System.out.println(e);
                        }
                }
        }
        System.out.println("Exit from C");
    }
}
class Mythread
{
    public static void main(String args[])

    {
        A  objA=new A( );
```

```
            B  objB=new B( );
            C  objC=new C( );
            System.out.println("Start Thread A");
            objA.start( );
            System.out.println("Start Thread B");
            objB.start( );
            System.out.println("Start Thread C");
            objC.start( );
            System.out.println("End of Main Thread");
        }
}
```

OUTPUT

```
Start Thread A
Start Thread B
Start Thread C
From Thread A: i= 1
Exit Thread B: j= 1
End of Main Thread

Exit from B
Exit Thread B: j= 2
```

Exit from B

From Thread A: i= 2

From Thread A: i= 3

From Thread A: i= 4

Exit Thread B: j= 3

From Thread A: i= 5

Exit from A

From thread C: k= 1

From thread C: k= 2

From thread C: k= 3

From thread C: k= 4

From thread C: k= 5

Exit from C

10.

AIM

Demonstration of fileinputstream and fileoutputstream classes

```
import java.io.FileInputStream;
import java.io.FileOutputStream;

public class InputOutputStream {
   public static void main(String[] args){
       Try
        {
          FileOutputStream fout=new
FileOutputStream("D:\\testout.txt");
          String s="Welcome to java";
          byte b[]=s.getBytes();//converting string into byte array
          fout.write(b);
          fout.close();
          System.out.println("Write success...");
       }
        catch(Exception e)
        {
          System.out.println(e);
       }
       Try
        {
          System.out.println("Reading file..");

                FileInputStream fin=new
FileInputStream("D:\\testout.txt");
                int i=0;
```

```
                while((i=fin.read())!=-1)
        {
                System.out.print((char)i);
         }
           fin.close();
        }
        catch(Exception e)
        {
           System.out.println(e);
        }
    }
}
```

OUTPUT

SHELL SCRIPT
AND
LINUX ADMINISTRATION

SHELL SCRIPTING

Program 1.
Get a name and number from the user, create a file with that name and number. Also display the contents of the file.
◦ If the name is XXX and number is 2 the filename must be XXX_2 ◦ use cat command to create a file
◦ Create the file with 10 different lines, then display the first 5 lines of file using head command.

Program

```
echo enter a name
read name

echo enter a number
read num
name+=_$num


echo enter 10 lines to the file $name and
save by pressing Ctrl+d cat > $name

echo -e "\n\nThe first 5 lines of the file $name:\n"
head -n 5 $name
```

Program 2

Write a program to greet a user by 'Good Morning', Good Afternoon' or 'Good Evening' based on time
◦ get the system time using 'date' command
◦ Read the name from the user
◦ if the name is 'XXX' then greet with 'Hello

XXX, Good Morning! ' **Program**

```
h=`date +%H`
echo Enter your name:
read name

if [ $h -le 12 ]
then
 echo Hello $name !! Good Morning !!
elif [ $h -le 16 ]
then
 echo Hello $name !! Good Afternoon !!
else
 echo Hello $name !! Good Evening !!
fi
```

Program 3.

Write a shell program to check whether a number is

positive,negative or zero **Program**

```
echo Enter a number
read n

if [ $n -eq 0 ]
then
 echo The number is zero.
elif [ $n -lt 0 ]
then
 echo The number a Negative Number.
else
 echo The number is a Positive Number.
fi
```

Program 4.

Shell Script To Print A Number In Reverse Order.

Program

```
echo Enter a number
read n
d=0
while [ $n -gt 0 ]
do
 k=`expr $n % 10`
 d=`expr $d \* 10 + $k`
 n=`expr $n / 10`
done

echo reversed number = $d
```

Program 5.

Write a program to check whether a user has logged in or not. The username is passed as command line argument.

Program

```
if [ $# -lt 1 ]
then
 echo "Improper usage! Correct usage is : $0 username"
 exit
fi
logname=$1
time=0
while true
do
 who | grep "$logname" > /dev/null
 if [ $? = 0 ]
 then
 echo $logname has logged in.
 if [ $time -ne 0 ]
 then
 echo He is $time minutes late.
 fi
 exit
 else
 time=`expr $time + 1`
 sleep 60
 fi
done
```

Program 6.

Write a demo program for the number and string comparison operators
◦ verfiy whether the entered username and password is of admin user's. if so display a warning message 'Permission denied'
◦ read a number from the user. Check whether number of files in a folder is greater than the read number.

Program

```
echo Enter username and password:
read uname
read pass

if [ $uname == "admin" ]
then
 echo Permission denied!
else
 echo enter the path to a folder
 read path
 set `ls $path`
 echo Enter a number:
 read n
 if [ $# -gt $n ]
 then
 echo Total number of files in the specified folder is
greater than $n  echo It is $#
 fi
fi
```

Program 7.
Write a demo program using basic calculator ◦ find the average size of the files available in a folder

Program

```
echo Enter path to a folder:
read path
cd $path

#Get the total size of files in bytes
set `wc -c * | tail -n 1`
a=$1


#Get the total number of files
excluding directories for i in `ls
$path`
do
 if [ -f $i ]
 then
 n=$(( $n + 1 ))
 fi
done

#Calculate average file size and print
avg=`echo "scale=2; $a / $n" | bc`
echo Total number of files : $n
echo Total file size: $a
echo The average size of files : $avg bytes
```

Program 8.

A program to create 10 users
◦ use loop structure
◦ get the usernames from the user
◦ assign same password to all the users

Program

```
for (( i=1; i<=10; i++ ))
do
 echo Enter a user name:
 read uname

 sudo useradd $uname
--password "redhat" done
```

Program 9.

A demo program to test
different file operators ◦ read
filename from the user
▪ Check if the file exists, if exists then display the contents, otherwise create the file
▪ Check whether the size of the file is zero
▪ check whether the file is having read, write and execute permission

Program

```
echo enter a file name
read fname

if [ -f $fname ]
then
 echo -e "File $fname exist and the content of the file:\n"
 cat < $fname
else
 echo The the file does not exist. Creating a new file.
 echo Enter the contents of the file $fname: and press Ctrl+d
 cat > $fname
fi


if [ -s $fname ]
then
 echo -e "\n\nThe size of the file is greater than zero."
else
 echo -e "\n\nThe size of the file is zero."
fi

if [ -r $fname -a -w $fname -a -x $fname ]
then
```

```
 echo You have read, write and execute

 permission to file $fname else
 echo You may not have read, write or execute permission.
fi
```

Program 10.

Write a program with 3 different functions. Use Menu driven program and invoke the function accrodingly
◦ Function for listing the contents of a folder
◦ Function for checking whether a file is available in a folder or not if so display the contents ◦ Function to check whether an user is already a member of a group

Program

```
listing() {
 echo Enter path to a folder
 read path

 if [ -d $path ]
 then
 echo -e "Contents of the directory $path :\n\n"
 ls $path
 else
 echo No such directory exists.
 fi
}

display() {
 echo Enter a file name:
 read fname
 if [ -f $fname ]
 then
 echo -e "The file $fname exists. The contents of the file is :\n\n"
 cat < $fname
 else
 echo File $fname does not exist.
 fi
}
```

```
groupcheck()
{
 echo Enter user name
 read uname
 echo enter group name
 read gname
 getent group $gname | grep $uname > /dev/null

 if [ $? -eq 0 ]
 then
 echo $uname is a member of $gname
 else
 echo $uname is not a member of $gname

 fi
 }


 while [ true ]
 do
 echo -e " \n\nMenu \n 1. List the contents of a direcrory \n 2. Display
the contents of a file \n  3. Check whether user belongs to a group\n
4. Exit"
 echo Enter your choice
 read ch

 case $ch in
 1) listing
 ;;
 2) display
 ;;
 3) groupcheck
 ;;
 4) exit
 ;;
 *) echo Enter a valid choice
 ;;  esac
```

done

1.
Adding and deleting user accounts, changing passwords

a) adduser
user@user-Inspiron-3268:~$ sudo su
[sudo] password for user:

root@user-Inspiron-3268:/home/user# adduser ashi
Adding user `ashi' ...
Adding new group `ashi' (1001) ...
Adding new user `ashi' (1001) with group `ashi' ...
The home directory `/home/ashi' already exists. Not copying from `/etc/skel'.
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for ashi
Enter the new value, or press ENTER for the default
Full Name []: as
Room Number []: 12
Work Phone []: 4567
Home Phone []: 8765
Other []:
Is the information correct? [Y/n] y

b)Changing password for user
root@user-Inspiron-3268:/home/user# passwd ashi
Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully

c)Deleting user
root@user-Inspiron-3268:/home/user# userdel ashi
root@user-Inspiron-3268:/home/user#

2.
Service command

a)Start a service
[fail]
65
user@user-Inspiron-3268:~$ sudo su
root@user-Inspiron-3268:/home/user# service postgresql start
* Starting PostgreSQL 9.3 database server

b)Stop a service
root@user-Inspiron-3268:/home/user# service postgresql stop
* Stopping PostgreSQL 9.3 database server [OK]

c)Restart a service
root@user-Inspiron-3268:/home/user# service postgresql restart
* Restarting PostgreSQL 9.3 database server [OK]

3.
Managing process

a)Listing of running process
user@user-Inspiron-3268:~$ ps
PID TTY TIME CMD
2813 pts/0 00:00:00 bash
2832 pts/0 00:00:00 ps

b) Full information of all running process
user@user-Inspiron-3268:~$ ps -f
UID PID PPID C STIME TTY TIME CMD
user 2813 2803 0 14:08 pts/0 00:00:00 bash
user 2843 2813 0 14:11 pts/0 00:00:00 ps -f

c) stopping a process
user@user-Inspiron-3268:~$ ps
PID TTY TIME CMD
3599 pts/9 00:00:00 bash
3615 pts/9 00:00:00 ps
user@user-Inspiron-3268:~$ ps -f
UID PID PPID C STIME TTY TIME CMD
user 3599 3054 0 15:34 pts/9 00:00:00 bash
user 3619 3599 0 15:35 pts/9 00:00:00 ps -f
user@user-Inspiron-3268:~$ kill 3599
66

4.
Setting the environmental variables

a) To setting a user defined variable
initializing variable
user@user-Inspiron-3268:~$ test_var='BCA'
displaying that variable
user@user-Inspiron-3268:~$ echo $test_var
BCA
setting the variable using export command
user@user-Inspiron-3268:~$ export test_var
print that varable only
user@user-Inspiron-3268:~$ printenv | grep test_var
test_var=BCA
changing the environmental variable like PATH
user@user-Inspiron-3268:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/ga
mes
user@user-Inspiron-3268:~$ export PATH=/home/user/BCA:$PATH
user@user-Inspiron-3268:~$ echo $PATH
/home/user/BCA:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/g
ames:/usr/local/games

5.

67

Sceduling job using cron

```
user@user-Inspiron-3268:~$ vi hello.txt
user@user-Inspiron-3268:~$ crontab -e
crontab: installing new crontab
user@user-Inspiron-3268:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and
cron(8)


#
# m h dom mon dow command
```

* * * * * echo "hello...." >> /home/user/hello.txt
user@user-Inspiron-3268:~$ cat hello.txt

hello....
hello....
hello....

6.
Setting the value of umask,changing the permission,changing owner and groups

a) Setting the value of umask

```
user@user-Inspiron-3268:~$ umask
0002
68
user@user-Inspiron-3268:~$ touch file1
user@user-Inspiron-3268:~$ ls -l file1
-rw-rw-r-- 1 user user 0 Jun 6 14:59 file1
user@user-Inspiron-3268:~$ umask 022
user@user-Inspiron-3268:~$ umask
0022
user@user-Inspiron-3268:~$ touch file2
user@user-Inspiron-3268:~$ ls -l file2
-rw-r--r-- 1 user user 0 Jun 6 15:00 file2
```

b)changing the permission

```
user@user-Inspiron-3268:~$ touch file1
user@user-Inspiron-3268:~$ ls -l file1
-rw-rw-r-- 1 user user 0 Jun 6 14:59 file1
user@user-Inspiron-3268:~$ chmod 777 file1
user@user-Inspiron-3268:~$ ls -l file1
-rwxrwxrwx 1 user user 0 Jun 6 14:59 file1
user@user-Inspiron-3268:~$ touch file2
user@user-Inspiron-3268:~$ ls -l file2
-rw-r--r-- 1 user user 0 Jun 6 15:00 file2
user@user-Inspiron-3268:~$ chmod +x file2


user@user-Inspiron-3268:~$ ls -l file2



-rwxr-xr-x 1 user user 0 Jun 6 15:00 file2
```

```
user@user-Inspiron-3268:~$ touch sample.txt
user@user-Inspiron-3268:~$ ls -l sample.txt
-rw-rw-r-- 1 user user 0 Jun 6 15:06 sample.txt


user@user-Inspiron-3268:~$ sudo chown user1:user1 sample.txt
user@user-Inspiron-3268:~$ ls -l sample.txt
-rw-rw-r-- 1 user1 user1 0 Jun 6 15:06 sample.txt
```

7.
Compressing and uncompressing files using anyone tool

a)Creating a file called BCA and displaying the details
user@user-Inspiron-3268:~$ vi bca
user@user-Inspiron-3268:~$ ls -l bca-rw-rw-r-- 1 user user 0 Jun 6 15:12 bca

b)Compressing file BCA using bzip2 command and displaying the details
69
user@user-Inspiron-3268:~$ bzip2 bca
user@user-Inspiron-3268:~$ ls -l bca.bz2
-rw-rw-r-- 1 user user 14 Jun 6 15:12 bca.bz2

c)Uncompressing the file BCA using the command bunzip2
user@user-Inspiron-3268:~$ bunzip2 bca.bz2
user@user-Inspiron-3268:~$ ls -l bca
-rw-rw-r-- 1 user user 0 Jun 6 15:12 bca

8.
Managing runlevel

a)Showing current runlevel
root@user-Inspiron-3268:/home/user# runlevel
N 2

b)Shutdown
#init 0

c)Reboot
#init 6