

SECTION – 4

Java Programming Lab

Session 1 : Data Types, Variables & Operators

Ex 1: Write a program in Java to implement the formula (Area = Height ×Width) to find the area of a rectangle. Where Height and Width are the rectangle's height and width.

Code:

```
class rectangle
{
    int h,w;
    rectangle(int x,int y)
    {
        h=x; w=y;
    }
    int area()
    {
        return(h*w);
    }
}
class s01_01
{
    public static void main(String args[])
    {
        rectangle r=new rectangle(10,20);
        int area=r.area();
        System.out.println("Area of Rectangle="+area);
    }
}
```

Ex 2: Write a program in Java to find the result of following expression (Assume a = 10, b = 5)

- i) $(a < 2) + (b > 2)$
- ii) $(a) \parallel (b > 0)$
- iii) $(a + b * 100) / 10$
- iv) $a \& b$

Code:

```
class s01_02
{
    public static void main(String args[])
    {
        int a=10,b=5;
        System.out.println(" i) (a<2)+(b>2): "+(a<2)+(b>2) );
        System.out.println(" ii) (a)||(b>0) : "+(a)||(b>0) );
        System.out.println("iii) (a+b*100)/10 : "+(a+b*100)/10 );
        System.out.println(" iv) (a&b) : "+(a&b) );
    }
}
```

Ex 3: Write a program in Java to explain the use of break and continue statements.

Code:

```
class s01_03
{
    public static void main(String args[])
    {
        int i = 0;
        int j = 0;
        for(int k=0; k<10; k++)
        {
            if(k%2==0) break;
            int l;
            try
            {
                l = 1 / k;
            }
            catch (Exception e)
            {
                continue;
            }
        }
    }
}
```

```

        l = Integer.parseInt(args[k]);
    }
    catch(NumberFormatException e)
    {
        j++;
        System.out.println("INVALID ARGUMENT :" + args[k]);
        continue;
    }

    i++;
    System.out.println("VALID ARGUMENT :" + args[k]);
}
System.out.println("NUMBER OF VALID ARGUMENTS :" + i);
System.out.println("NUMBER OF INVALID ARGUMENT:" + j);
}
}

```

Ex 4: Write a program in Java to find the average of marks you obtained in your 10+2 class.

Code:

```

class s01_04
{
    public static void main(String args[])
    {
        int reg,m1,m2,m3;
        String name;
        reg=Integer.parseInt(args[0]);
        name=args[1];
        m1=Integer.parseInt(args[2]);
        m2=Integer.parseInt(args[3]);
        m3=Integer.parseInt(args[4]);
        System.out.println("-----");
        System.out.println(" MARK LIST ");
        System.out.println("-----");
        System.out.println("Register No : "+reg);
        System.out.println("Name : "+name);
        System.out.println("Mark1 : "+m1);
        System.out.println("Mark2 : "+m2);
        System.out.println("Mark3 : "+m3);
        System.out.println("Total : "+(m1+m2+m3));
        System.out.println("Average : "+(float)(m1+m2+m3)/3);
        System.out.println("-----");
    }
}

```

Session 2 : Statements and Array

Ex 5: Write a program in Java to find $A \times B$ where A is a matrix of 3×3 and B is a matrix of 3×4 . Take the values in matrixes A and B from the user.

Code:

```

class s02_01
{
    public static void main(String arg[])
    {
        int a[][]=new int[3][3];
        int b[][]=new int[3][3];
        int c[][]=new int[3][3];
        int n=0;
        for(int i=0;i<3;i++)
        for(int j=0;j<3;j++)
        a[i][j]=Integer.parseInt(arg[n++]);
        for(int i=0;i<3;i++)
        for(int j=0;j<3;j++)
        b[i][j]=Integer.parseInt(arg[n++]);
        //multiplying the two matrix
        for(int i=0;i<3;i++)

```

```

        for(int j=0;j<3;j++)
        {
            c[i][j]=0;
            for(int k=0;k<3;k++)
                c[i][j]+=(a[i][k]*b[k][j]);
        }
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
                System.out.print(c[i][j]+" ");
            System.out.println();
        }
    }
}

```

Ex 6: Write a program in Java to compute the sum of the digits of a given integer. Remember, your integer should not be less than the five digits.

Code:

```

class s02_02
{
    public static void main(String arg[])
    {
        int sum=0;
        long n=Long.parseLong(arg[0]);
        while(n>0)
        {
            sum+=n%10;
            n/=10;
        }
        System.out.println("Sum="+sum);
    }
}

```

Session 3 : Class And Objects

Ex 7: Write a program in Java with class Rectangle with the data fields width, length, area and colour. The length, width and area are of double type and colour is of string type. The methods are set_length (), set_width (), set_colour(), and find_area ().

Code:

```

import java.io.*;
class rect
{
    int width,length;
    String color;
    void set_length(int a)
        { length=a; }
    void set_width(int a)
        { width=a; }
    void set_color(String a)
        { color=a; }
    int area()
        { return(width*length); }
    String getcolor()
        { return(color); }
}
class s03_01
{
    public static void main(String arg[])throws Exception
    {
        String s=null;
        DataInputStream in=new DataInputStream(System.in);
        rect a=new rect();
        System.out.println("Enter the length for first rectangle");
        s=in.readLine();
    }
}

```

```

        a.set_length(Integer.parseInt(s));
        System.out.println("Enter the width for first rectangle");
        s=in.readLine();
        a.set_width(Integer.parseInt(s));
        System.out.println("Enter the Color for first rectangle");
        a.set_color(in.readLine());
        rect b=new rect();
        System.out.println("Enter the length for second rectangle");
        s=in.readLine();
        b.set_length(Integer.parseInt(s));
        System.out.println("Enter the width for second rectangle");
        s=in.readLine();
        b.set_width(Integer.parseInt(s));
        System.out.println("Enter the Color for second rectangle");
        b.set_color(in.readLine());
        if(a.area()==b.area() && a.getcolor().equals(b.getcolor()))
        System.out.println("Matching Rectangle ");
        else
        System.out.println("Non Matching Rectangle ");
    }
}

```

Ex 8: Create a class Account with two overloaded constructors. The first constructor is used for initializing, the name of account holder, the account number and the initial amount in the account. The second constructor is used for initializing the name of the account holder, the account number, the addresses, the type of account and the current balance. The Account class is having methods Deposit (), Withdraw (), and Get_Balance(). Make the necessary assumption for data members and return types of the methods. Create objects of Account class and use them.

Code:

```

class account
{
    String name,address,type;
    int accno,bal;
    account(String n,int no,int b)
    { name=n; accno=no; bal=b; }
    account(String n,int no,String addr,String t,int b)
    {
        name=n; accno=no;
        address=addr;
        type=t; bal=b;
    }
    void deposit(int a) { bal+=a; }
    void withdraw(int a) { bal-=a; }
    int getbalance() { return(bal); }
    void show()
    {
        System.out.println("_____");
        System.out.println(" ACCOUNT DETAILS");
        System.out.println("-----");
        System.out.println("Name : "+name);
        System.out.println("Account No : "+accno);
        System.out.println("Address : "+address);
        System.out.println("Type : "+type);
        System.out.println("Balance : "+bal);
        System.out.println("-----");
    }
}
class s03_02
{
    public static void main(String arg[])throws Exception
    {
        account a1=new account("Anil",555,5000);
        account a2=new account("Anil",666,"Tirur","Current account",1000);
        a1.address="Calicut";
    }
}

```

```

        a1.type="fixed deposit";
        a1.deposit(5000);
        a2.withdraw(350);
        a2.deposit(a2.getbalance());
        a1.show();
        a2.show();
    }
}

```

Ex 9: Write a program in Java to create a stack class of variable size with push() and pop () methods. Create two objects of stack with 10 data items in both. Compare the top elements of both stack and print the comparison result.

Code:

```

import java.io.*;
class stack
{
    int data[]=new int[50];
    int sp=0;
    int pop()
    {
        if(sp<=0)
        {
            System.out.println("Stack is empty");
            return(0);
        }
        else
            return(data[sp--]);
    }
    void push(int a)
    {
        if(sp>=50)
            System.out.println("Stack overflow");
        else
            data[sp++]=a;
    }
}
class s03_03
{
    public static void main(String arg[])
    {
        DataInputStream in=null;
        String s;
        int d;
        stack s1=new stack();
        stack s2=new stack();
        try
        {
            in=new DataInputStream(System.in);
            for(int i=0;i<10;i++)
            {
                System.out.println(1+i+" Enter data for the first stack");
                s=in.readLine();
                d=Integer.parseInt(s);
                s1.push(d);
            }
            for(int i=0;i<10;i++)
            {
                System.out.println(1+i+" Enter data for the second stack");
                s=in.readLine();
                d=Integer.parseInt(s);
                s2.push(d);
            }
            if(s1.pop()==s2.pop())
                System.out.println("The top of the stacks are same");
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}

```

```

        else
        System.out.println("The top of the stacks are same");
    }
    catch(Exception e) { System.out.println(e); }
}

```

Session 4 : Inheritance and Polymorphism

Ex 10: Write a Java program to show that private member of a super class cannot be accessed from derived classes.

Code:

```

class room
{
    private int l,b;
    room(int x,int y)
    { l=x; b=y;}
    int area()
    {return(l*b);}
}
class class_room extends room
{
    int h;
    class_room(int x,int y,int z)
    {
        super(x,y);
        h=z;
    }
    int volume()
    {
        return(area()*h);
    }
}
class s04_01
{
    public static void main(String args[])
    {
        class_room cr=new class_room(10,20,15);
        int a1=cr.area();
        int v1=cr.volume();
        System.out.println("Area of Room : "+a1);
        System.out.println("Volume of Room : "+v1);
    }
}

```

Ex 11: Write a program in Java to create a Player class. Inherit the classes Cricket _Player, Football _Player and Hockey_ Player from Player class.

Code:

```

class player
{
    String name;
    int age;
    player(String n,int a)
    { name=n; age=a; }
    void show()
    {
        System.out.println("\n");
        System.out.println("Player name : "+name);
        System.out.println("Age : "+age);
    }
}
class cricket_player extends player
{
    String type;
}

```

```

        cricket_player(String n,String t,int a)
        {
            super(n,a);
            type=t;
        }
        public void show()
        {
            super.show();
            System.out.println("Player type : "+type);
        }
    }
    class football_player extends player
    {
        String type;
        football_player(String n,String t,int a)
        {
            super(n,a);
            type=t;
        }
        public void show()
        {
            super.show();
            System.out.println("Player type : "+type);
        }
    }
    class hockey_player extends player
    {
        String type;
        hockey_player(String n,String t,int a)
        {
            super(n,a);
            type=t;
        }
        public void show()
        {
            super.show();
            System.out.println("Player type : "+type);
        }
    }
    //----- main -----
    class s04_02
    {
        public static void main(String args[])
        {
            cricket_player c=new cricket_player("Ameer","cricket",25);
            football_player f=new football_player("arun","foot ball",25);
            hockey_player h=new hockey_player("Ram","hockey",25);
            c.show();
            f.show();
            h.show();
        }
    }
}

```

Ex 12: Write a class Worker and derive classes DailyWorker and SalariedWorker from it. Every worker has a name and a salary rate. Write method ComPay (int hours) to compute the week pay of every worker. A Daily Worker is paid on the basis of the number of days s/he works. The Salaried Worker gets paid the wage for 40 hours a week no matter what the actual hours are. Test this program to calculate the pay of workers. You are expected to use the concept of polymorphism to write this program.

Code:

```

class worker
{
    String name;
    int empno;
    worker(int no,String n)

```

```

        { empno=no; name=n; }
void show()
{
    System.out.println("\n-----");
    System.out.println("Employee number : "+empno);
    System.out.println("Employee name : "+name);
}
}
class dailyworker extends worker
{
    int rate;
    dailyworker(int no,String n,int r)
    {
        super(no,n);
        rate=r;
    }
    void compay(int h)
    {
        show();
        System.out.println("Salary : "+rate*h);
    }
}
class salariedworker extends worker
{
    int rate;
    salariedworker(int no,String n,int r)
    {
        super(no,n);
        rate=r;
    }
    int hour=40;
    void compay()
    {
        show();
        System.out.println("Salary : "+rate*hour);
    }
}
//----- main -----
class s04_03
{
    public static void main(String args[])
    {
        dailyworker d=new dailyworker(254,"Arjun",75);
        salariedworker s=new salariedworker(666,"Unni",100);
        d.compay(45);
        s.compay();
    }
}

```

Ex 13: Consider the trunk calls of a telephone exchange. A trunk call can be ordinary, urgent or lightning. The charges depend on the duration and the type of the call. Write a program using the concept of polymorphism in Java to calculate the charges.

Code:

```

import java.io.*;
class call
{
    float dur;
    String type;
    float rate()
    {
        if(type.equals("urgent"))
            return 4.5f;
        else if(type=="lightning")
            return 3.5f;
    }
}

```



```

        else
        return 3f;
    }
}
class bill extends call
{
    float amount;
    DataInputStream in=null;
    bill()
    {
        try
        {
            in=new DataInputStream(System.in);
        }
        catch(Exception e)
        { System.out.println(e); }
    }
    void read()throws Exception
    {
        String s;
        System.out.println("enter call type(urgent,lightning,ordinary):");
        type=in.readLine();
        System.out.println("enter call duration:");
        s=in.readLine();
        dur=Float.valueOf(s).floatValue();
    }
    void calculate()
    {
        if(dur<=1.5)
            amount=rate()*dur+1.5f;
        else if(dur<=3)
            amount=rate()*dur+2.5f;
        else if(dur<=5)
            amount=rate()*dur+4.5f;
        else
            amount=rate()*dur+5f;
    }
    void print()
    {
        System.out.println("*****");
        System.out.println(" PHONE BILL ");
        System.out.println("*****");
        System.out.println(" Call type : "+type);
        System.out.println(" Duration : "+dur);
        System.out.println(" CHARGE : "+amount);
        System.out.println("*****");
    }
}
}
class s04_04
{
    public static void main(String arg[])throws Exception
    {
        bill b=new bill();
        b.read();
        b.calculate();
        b.print();
    }
}

```

Session 5 : Package and Interface

Ex 14: Write a program to make a package Balance in which has Account class with Display_Balance method in it. Import Balance package in another program to access Display_Balance method of Account class.

Code:

```

class s05_01
{
    public static void main(String ar[])
    {
        try
        {
            balance.account a=new balance.account();
            a.read();
            a.disp();
        }
        catch(Exception e)
        { System.out.println(e); }
    }
}
package balance;
import java.io.*;
public class account
{
    long acc,bal;
    String name;
    public void read()throws Exception
    {
        DataInputStream in=new DataInputStream(System.in);
        System.out.println("Enter the name :");
        name=in.readLine();
        System.out.println("Enter the account number :");
        acc=Long.parseLong(in.readLine());
        System.out.println("Enter the account balance :");
        bal=Long.parseLong(in.readLine());
    }
    public void disp()
    {
        System.out.println("~~~~~");
        System.out.println("--- Account Details ---");
        System.out.println("~~~~~");
        System.out.println("Name :"+name);
        System.out.println("Account number :"+acc);
        System.out.println("Balance :"+bal);
    }
}

```

Ex 15: Write a program in Java to show the usefulness of Interfaces as a place to keep constant value of the program.

Code:

```

interface area
{
    static final float pi=3.142f;
    float compute(float x,float y);
}
class rectangle implements area
{
    public float compute(float x,float y)
    {return(x*y);}
}
class circle implements area
{
    public float compute(float x,float y)
    {return(pi*x*x);}
}
class s05_02
{
    public static void main(String args[])
    {
        rectangle rect=new rectangle();
        circle cr=new circle();
    }
}

```

```

        area ar;
        ar=rect;
        System.out.println("Area of the rectangle= "+ar.compute(10,20));
        ar=cr;
        System.out.println("Area of the circle= "+ar.compute(10,0));
    }
}

```

Ex 16: Create an Interface having two methods division and modules. Create a class, which overrides these methods.

Code:

```

interface course
{
    void division(int a);
    void modules(int b);
}
class stud implements course
{
    String name;
    int div,mod;
    void name(String n)
    { name=n; }
    public void division(int a)
    { div=a; }
    public void modules(int b)
    { mod=b; }
    void disp()
    {
        System.out.println("Name :"+name);
        System.out.println("Division :"+div);
        System.out.println("Modules :"+mod);
    }
}
//-----main-----
class s05_03
{
    public static void main(String args[])
    { stud s=new stud();
      s.name("Arun");
      s.division(5);
      s.modules(15);
      s.disp();
    }
}

```

Session 6 : Exception Handling

Ex 18: Write a program in Java to display the names and roll numbers of students. Initialize respective array variables for 10 students. Handle Array Index Out Of Bounds Exeption, so that any such problem doesn't cause illegal termination of program.

Code:

```

import java.io.*;
class student
{
    String name,grade;
    int reg,m1,m2,m3;
    void read()throws Exception
    {
        DataInputStream in= new DataInputStream(System.in);
        System.out.println("enter the register no : ");
        reg=Integer.parseInt(in.readLine());
        System.out.println("enter the name : ");
        name=in.readLine();
        System.out.println("enter mark1 : ");
    }
}

```

```

        m1=Integer.parseInt(in.readLine());
        System.out.println("enter mark2 : ");
        m2=Integer.parseInt(in.readLine());
        System.out.println("enter mark3 : ");
        m3=Integer.parseInt(in.readLine());
    }
    public void disp_grade()
    {
        int tt=m1+m2+m3;
        if(tt>=250) grade="A";
        else if(tt>=200) grade="B";
        else if(tt>=150) grade="C";
        else if(tt>=100) grade="D";
        else grade="E";
        System.out.println("Grade :"+grade);
    }
    void disp()
    {
        System.out.println("~~~~~");
        System.out.println(" MARK LIST OF STUDENTS ");
        System.out.println("~~~~~");
        System.out.println("Register No :"+reg);
        System.out.println("Name :"+name);
        System.out.println("Mark1 :"+m1);
        System.out.println("Mark2 :"+m2);
        System.out.println("Mark3 :"+m3);
        disp_grade();
    }
}
class s06_01
{
    public static void main(String ar[])
    {
        int no=0;
        student s=new student();
        try
        {
            DataInputStream in= new DataInputStream(System.in);
            System.out.println("enter the number of students : ");
            no=Integer.parseInt(in.readLine());
            for(int i=0;i<no;i++);
            s.read();
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("the maximum students should be ten\n");
            no=10;
        }
        catch(Exception e)
        { System.out.println(e); }
        for(int i=0;i<no;i++);
        s.disp();
    }
}

```

Ex 19: Write a Java program to enable the user to handle any chance of divide by zero exception.

Code:

```

class s06_02
{
    public static void main(String ar[])
    {
        int no=0,m=10,result=0;
        try
        {
            result=m/no;

```

```

    }
    catch(ArithmeticException e)
    {
        System.out.println(" division by zero ");
        System.out.println(" value of result has been set as one");
        result=1;
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    System.out.println("Result :"+result);
}
}

```

Ex 20: Create an exception class, which throws an exception if operand is nonnumeric in calculating modules. (Use command line arguments).

Code:

```

class NonNum extends Exception
{
    NonNum()
    { super("the value is non numeric \n"); }
}
class s06_03
{
    public static void main(String ar[])
    {
        int a,b,c=0;
        try
        {
            a=Integer.parseInt(ar[0]);
            throw new NonNum();
        }
        catch(NumberFormatException e)
        {System.out.println(e);}
        catch(NonNum e)
        { System.out.println(e);}
    }
}

```

Ex 21: On a single track two vehicles are running. As vehicles are going in same direction there is no problem. If the vehicles are running in different direction there is a chance of collision. To avoid collisions write a Java program using exception handling. You are free to make necessary assumptions.

Code:

```

import java.io.*;
class collision extends Exception
{
    collision(String s)
    { super(s); }
}
class s06_04
{
    public static void main(String ar[])
    {
        String t1=null,t2=null;
        try
        {
            DataInputStream in= new DataInputStream(System.in);
            System.out.println("enter the direction of vehicle1:(left/right):");
            t1=in.readLine();
            System.out.println("enter the direction of vehicle2:(left/right):");
            t2=in.readLine();
            if(!t1.equals(t2))
                throw new collision("truck2 has to go on "+ t1 +" direction");
        }
    }
}

```

```

    }
    catch(collision e)
    {
        System.out.println(e);
        t2=t1;
        System.out.println("the collision has been avoided by redirection
        truck2");
    }
    catch(Exception e)
    { System.out.println(e); }
    System.out.println("direction of truck1 :"+t1);
    System.out.println("direction of truck2 :"+t2);
}
}

```

Session 7 : Multithreading

Ex 23: Write a program to launch 10 threads. Each thread increments a counter variable. Run the program with synchronization.

Code:

```

class s07_02
{
    public static void main(String arg[])throws Exception
    {
        data d1=new data();
        data d2=new data();
        data d3=new data();
        data d4=new data();
        data d5=new data();
        data d6=new data();
        data d7=new data();
        data d8=new data();
        data d9=new data();
        data d10=new data();
        System.out.println(d10.count);
    }
}
//-----
class item { static int count=0; }
class data extends item implements Runnable
{
    item d=this;
    Thread t;
    data()
    {
        t=new Thread(this);
        t.start();
    }
    public void run()
    { d=syn.increment(d); }
}
//=====
class syn
{
    synchronized static item increment(item i)
    {
        i.count++;
        return(i);
    }
}

```

Ex 24: Write a program for generating 2 threads, one for printing even numbers and the other for printing odd numbers.

Code:

```

class even extends Thread
{
    Thread t=null;
    even()
    {
        t=new Thread(this);
        start();
    }
    public void run()
    {
        try
        {
            for(int i=2;i<50;i+=2)
                System.out.print(i+" ");
            Thread.sleep(100);
        }
        catch(Exception e)
        {System.out.println("thread interepted");}
    }
}
class odd extends Thread
{
    Thread t=null;
    odd()
    {
        t=new Thread(this);
        start();
    }
    public void run()
    {
        try
        {
            for(int i=1;i<50;i+=2)
                System.out.print(i+" ");
            Thread.sleep(100);
        }
        catch(Exception e)
        {System.out.println("thread interepted");}
    }
}
class s07_03
{
    public static void main(String arg[])
    {
        even e=new even();
        odd o=new odd();
    }
}

```

Session 8 : Reading, Writing & String Handling in Java

Ex 26: Writ a program in Java to create a String object. Initialize this object with your name. Find the length of your name using the appropriate String method. Find whether the character 'a' is in your name or not; if yes find the number of times 'a' appears in your name. Print locations of occurrences of 'a' .Try the same for different String objects.

Code:

```

class data
{
    String name;
    data(String n){ name=n; }
    void disp()
    {
        System.out.println("~~~~~");
        System.out.println("Name :"+name);
    }
}

```

```

        int c=0;
        int len=name.length();
        for(int i=0;i<len;i++)
        if(name.charAt(i)=='A' || name.charAt(i)=='a')
        {
            c++;
            System.out.println("number of occurance :"+c);
            System.out.println("Possition :"+(i+1));
        }
        if(c==0)
        System.out.println("there is no 'A' available in the string");
    }
}
class s08_01
{
    public static void main(String ar[])
    {
        {
            data d1=new data("anil kumar");
            d1.disp();
            data d2=new data("biju");
            d2.disp();
        }
    }
}

```

Ex 28: Write a program for searching strings for the first occurrence of a character or substring and for the last occurrence of a character or substring.

Code:

```

import java.io.*;
class s08_03
{
    public static void main(String[]args) throws Exception
    {
        int len1,len2,last=0;
        DataInputStream in=new DataInputStream(System.in);
        System.out.println("Enter the string:");
        String s1=in.readLine();
        System.out.println("Enter searching string:");
        String s2=in.readLine();
        len1=s1.length();
        len2=s2.length();
        for(int i=0;i<=(len1-len2);i++)
        {
            if(s1.substring(i,len2+i).equals(s2))
            {
                if(last==0)
                System.out.println("first occurance is at possition :"+(i+1));
                last=i+1;
            }
        }
        if(last!=0)
        System.out.println("last occurance is at position :"+last);
        else
        System.out.println("the string is not found");
    }
}

```

Ex 29: Write a program in Java to read a statement from console, convert it into upper case and again print on console.

Code:

```

import java.io.*;
class s08_04
{
    public static void main(String a[]) throws IOException
    {

```



```

        DataInputStream in=new DataInputStream(System.in);
        System.out.println("Enter file Statement:");
        String s1=in.readLine();
        System.out.println(s1.toUpperCase());
    }
}

```

Ex 30: Write a program in Java, which takes the name of a file from user, read the contents of the file and display it on the console.

Code:

```

import java.io.*;
class s08_05
{
    public static void main(String args[])
    {
        FileInputStream fis=null;
        Try
        {
            fis=new FileInputStream(args[0]);
            byte ch;
            while((ch=(byte)fis.read())!=-1)
                {System.out.print((char)ch); }
        }
        catch(IOException e)
        { System.out.println("interrupted");}
        finally
        {
            try
            {fis.close();}
            catch(IOException e)
            {System.out.println("error in closing");}
        }
    }
}

```

Ex 31: Write a Java program to copy a file into another file.

Code:

```

import java.io.*;
class s08_06
{
    public static void main(String a[]) throws IOException
    {
        int d=0;
        FileInputStream fi=null;
        FileOutputStream fo=null;
        DataInputStream in=new DataInputStream(System.in);
        System.out.println("Enter file name:");
        String s1=in.readLine();
        System.out.println("Enter new file name:");
        String s2=in.readLine();
        try
        {
            fi=new FileInputStream(s1);
            fo=new FileOutputStream(s2);
            System.out.println("Copying file");
            while((d=fi.read())!=-1)
            {
                fo.write((byte)d);
                System.out.println("File copied");
            }
        }
        catch(IOException e)
        {System.out.println(e);}
    }
}

```

Session 9 : Applets and its Application

Ex 32: Write a Java Applet program which reads your name and address in different text fields and when a button named find is pressed the sum of the length of characters in name and address is displayed in another text field. Use appropriate colors, layout to make your applet look good.

Code:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class s09_01 extends Applet implements ActionListener
{
    public void init()
    {
        label1 = new Label();
        label2 = new Label();
        label3 = new Label();
        t1 = new TextField();
        t2 = new TextField();
        t3 = new TextField();
        b1 = new Button();
        setLayout(null);
        setBackground(new Color(0, 153, 102));
        label1.setAlignment(Label.RIGHT);
        label1.setText("Name");
        add(label1);
        label1.setBounds(140, 60, 50, 20);
        label2.setAlignment(Label.RIGHT);
        label2.setText("Address");
        add(label2);
        label2.setBounds(140, 90, 50, 20);
        label3.setAlignment(Label.RIGHT);
        label3.setText("Total length");
        add(label3);
        label3.setBounds(130, 120, 60, 20);
        add(t1);
        t1.setBounds(200, 60, 100, 20);
        add(t2);
        t2.setBounds(200, 90, 100, 20);
        add(t3);
        t3.setBounds(200, 120, 100, 20);
        b1.setBackground(new Color(255, 204, 153));
        b1.setLabel("Total");
        b1.addActionListener(this);
        add(b1);
        b1.setBounds(150, 180, 80, 24);
    }
    public void actionPerformed(ActionEvent ae)
    {
        int a=t1.getText().length();
        a+=t2.getText().length();
        t3.setText(Integer.toString(a));
    }
    Label label1;
    Label label2;
    Label label3;
    TextField t1;
    TextField t2;
    TextField t3;
    Button b1;
}

<html>
<body>
<applet code="s09_01.class" width=400 height=400>
</applet>
```

```
</body>
</html>
```

Ex 33: Create an applet which displays a rectangle/string with specified color & coordinate passed as parameter from the HTML file.

Code:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class s09_02 extends Applet implements ActionListener
{
    public void init()
    {
        setLayout(null);
        setBackground(new Color(0,10,100));
    }
    public void paint(Graphics p)
    {
        String t=null;
        int x,y,w,h,r,g,b;
        t=getParameter("xx");
        x=Integer.parseInt(t);
        t=getParameter("yy");
        y=Integer.parseInt(t);
        t=getParameter("ww");
        w=Integer.parseInt(t);
        t=getParameter("hh");
        h=Integer.parseInt(t);
        t=getParameter("rr");
        r=Integer.parseInt(t);
        t=getParameter("gg");
        g=Integer.parseInt(t);
        t=getParameter("bb");
        b=Integer.parseInt(t);
        p.setColor(new Color(r,g,b));
        p.fillRect(x,y,w,h);
    }
}
```

```
<html>
<body>
<applet code="s09_02.class" width=400 height=400>
<param name="xx" value="25">
<param name="yy" value="25">
<param name="ww" value="150">
<param name="hh" value="150">
<param name="rr" value="0">
<param name="gg" value="150">
<param name="bb" value="100">
</applet>
</body>
</html>
```

Ex 34: Create an applet which will display the calendar of a given date.

Code:

```
import java.util.*;
import java.awt.*;
import java.applet.*;
public class s9e3 extends Applet
{
    GregorianCalendar cal=new GregorianCalendar();
    String s,s1,s2,s3,s4;
    int a=0,b=0,c=0,d=0;
    public void start()
```

```

    {
        s=getParameter("fg");
        s1=getParameter("as");
        s2=getParameter("as1");
        s3=getParameter("as2");
        s4=getParameter("as3");
        a=Integer.parseInt(s1);
        b=Integer.parseInt(s2);
        c=Integer.parseInt(s3);
        d=Integer.parseInt(s4);
    }
    public void paint(Graphics g)
    {
        if(s.equals("red"))
            g.setColor(Color.red);
        g.drawRect(a,b,c,d);
        g.drawString("Color = "+s,25,25);
        g.drawString("Calendar is "+cal.DATE+"/"+cal.MONTH+"/"+cal.YEAR,34,36);
    }
}

```

Ex 35: Write a program to store student's detail using Card Layout.

Code:

```

class s9e4 extends Applet
{
    CardLayout c1;
    Panel p;
    Label l1;
    Label l2;
    Label l3;
    Label l4;
    TextField t1;
    TextField t2;
    TextField t3;
    TextField t4;
    public void start()
    {
    }
    public void init()
    {
        c1=new CardLayout();
        l1=new Label("Enter Name :");
        l2=new Label("Enter Place :");
        l3=new Label("Address :mo(ho)");
        l4=new Label("Pin :670571 ");
        t1=new TextField(20);
        p=new Panel();
        p.setLayout(c1);
        add(l1);
        add(t1);
        add(l2);
        // add(t2);
        add(l3);
        // add(t3);
        add(l4);
        // add(t4);
    }
    public void paint(Graphics g)
    {
    }
}

```

Session 10 : Networking & Other Advanced Feature of JAVA

Ex 37: Write a Java program to find the numeric address of the following web sites

- i. www.ignou.ac.in
- ii. www.indiatimes.com
- iii. www.rediff.com
- iv. www.apple.com

In addition to this, find the Internet Address of your local host.

Code:

```
import java.net.*;
class s10_01
{
    public static void main(String ar[])throws Exception
    {
        InetAddress a=InetAddress.getLocalHost();
        System.out.println("The Local Host IP:"+a);
        a=InetAddress.getByName("www.ignou.ac.in");
        System.out.println("The IP of www.ignou.ac.in :"+a);
        a=InetAddress.getByName("www.indiatimes.com");
        System.out.println("The IP of www.indiatimes.com :"+a);
        a=InetAddress.getByName("www.apple.com");
        System.out.println("The IP of www.apple.com :"+a);
        InetAddress s[]=InetAddress.getAllByName("www.rediff.com");
        for(int i=0;i<s.length;i++)
            System.out.println("The IP of www.rediff.com :"+s[i]);
    }
}
```

Ex 38: Create an applet which takes name and age as parameters and display the message "<name> is <age> year old.". Print the URL of the class file.

Code:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class s10_02 extends Applet
{
    public void init()
    {
        Font f=new Font("adobe-courier",Font.PLAIN,18);
        setFont(f);
        setLayout(null);
        setForeground(Color.green);
        setBackground(new Color(200,0,200));
    }
    public void paint(Graphics g)
    {
        String n,a,msg;
        n=getParameter("name");
        a=getParameter("age");
        msg=n+" is "+a+" year old. ";
        g.drawString(msg,25,25);
    }
}

<html>
<body>
<applet code="s10_02.class" width=400 height=400>
<param name="name" value="Anil kumar">
<param name="age" value="25">
</applet>
</body></html>
```

ignousite.blogspot.com