

**DWARAKA DOSS GOVERDHAN DOSS VAISHNAV
EVENING COLLEGE (AUTONOMOUS)
ARUMBAKKAM, CHENNAI-600 106**



DEPARTMENT OF COMPUTER APPLICATIONS

Bachelor of Computer Applications

**PROGRAMMING IN JAVA
LAB**

**DWARAKA DOSS GOVERDHAN DOSS VAISHNAV EVENING
COLLEGE (AUTONOMOUS)
ARUMBAKKAM, CHENNAI-600 106**



**DEPARTMENT OF COMPUTER APPLICATIONS
PROGRAMMING IN JAVA LAB**

*Certified that this is a record work done by whose
Register no. of II B.C.A., during the academic year 2015 - 2016.*

Faculty In-charge

Head of the Department

*Submitted for practical examination held on at Dwaraka Doss
Goverdhan Doss Vaishnav College, Chennai-106.*

Internal Examiner

External Examiner

JAVA PROGRAMMING LAB

| Sl.No. | Date | Program Name | Page No | Signature |
|--------------------------|-------------|--------------------------------------|----------------|------------------|
| Java Applications | | | | |
| 1 | | Constructors | | |
| 2 | | String Class and its Methods | | |
| 3 | | StringBuffer Class and its Methods | | |
| 4 | | Random Class | | |
| 5 | | Vector Class | | |
| 6 | | Multilevel Inheritance | | |
| 7 | | Hierarchical Inheritance | | |
| 8 | | Multiple Inheritance using Interface | | |
| 9 | | Packages | | |
| 10 | | Built-in Exception Handling | | |
| 11 | | User-Defined Exception Handling | | |
| 12 | | Threads | | |
| 13 | | File Manipulation | | |
| Java Applets | | | | |
| 14 | | Working With Fonts And Colors | | |
| 15 | | Parameter Passing Technique | | |
| 16 | | Drawing various Shapes | | |
| 17 | | Temperature Conversion | | |
| 18 | | Simple Calculator | | |

EX. NO: 1
DATE:

CONSTRUCTORS

AIM:

To write a Java program to illustrate class, method, object and constructors.

ALGORITHM:

- Step1: Define a class "Student" with name, register number and 3 marks.
- Step 2: Declare three constructors – Default, Parameterized and Copy constructor to initialize the data members of the class.
- Step 3: In the display() method, print all the details.

PROGRAM:

```
class Student
{
    String name;
    int regno;
    int mark1,mark2,mark3;

    // Default constructor
    Student()
    {
        name="Raju";
        regno=12345;
        mark1=56;
        mark2=47;
        mark3=78;
    }

    // Parameterized Constructor
    Student(String name,int regno,int mark1,int mark2,int mark3)
    {
        this.name=name;
        this.regno=regno;
        this.mark1=mark1;
        this.mark2=mark2;
        this.mark3=mark3;
    }
}
```

```
//Copy Constructor
Student(Student s)
{
    name=s.name;
    regno=s.regno;
    mark1=s.mark1;
    mark2=s.mark2;
    mark3=s.mark3;
}
void display()
{
    System.out.println(name + "\t" +regno+ "\t" +mark1+ "\t" +mark2+ "\t" + mark3);
}
public static void main(String args[])
{
    Student s1=new Student();
    Student s2=new Student("Bala",34266,58,96,84);
    Student s3=new Student(s1);
    System.out.println("Name" + "\t" +"Regno"+ "\t" +"Marks1"+ "\t" +"Mark2"+ "\t" +
    "Mark3");
    s1.display();
    s2.display();
    s3.display();
}
```

EX. NO: 2

DATE:

STRING CLASS AND ITS METHODS

AIM:

To write a Java program using the methods of String class.

ALGORITHM:

- Step 1: Initialize strings using constructor and “=” operator.
- Step 2: Calculate the length of string using length() method.
- Step 3: Predict the occurrence of a character in the string using indexOf() method.
- Step 4: Convert the case of strings using toUpperCase() and toLowerCase() methods.
- Step 5: Check the equality of strings using equals() method.
- Step 6: Compare the strings using compareTo() method and display the result.

PROGRAM:

```
class Strings
{
    public static void main(String args[])
    {
        String s1=new String("Welcome to Java");
        String s2="WELCOME TO JAVA";
        System.out.println(" The string s1 is : " +s1);
        System.out.println(" The string s1 is : " +s2);
        System.out.println(" Length of the string s1 is : " +s1.length());
        System.out.println(" The first accurence of o is at the position : " +s1.indexOf('o'));
        System.out.println(" The String in Upper Case : " +s1.toUpperCase());
        System.out.println(" The String in Lower Case : " +s1.toLowerCase());
        System.out.println(" s1 equals to s2 : " +s1.equals(s2));
        System.out.println(" s1 equals ignore case to s2 : " +s1.equalsIgnoreCase(s2));
        int result=s1.compareTo(s2);
        System.out.println("After compareTo()");
        if(result==0)
            System.out.println( s1 + " is equal to "+s2);
        else
        if(result>0)
            System.out.println( s1 + " is greater than to "+s2);
        else
            System.out.println( s1 + " is smaller than to "+s2);
        System.out.println(" Character at an index of 6 is :" +s1.charAt(6));
        String s3=s1.substring(4,12);
        System.out.println(" Extracted substring is :" +s3);
        System.out.println(" After Replacing o with x in s1 : " +s1.replace('o','x'));
        String s4=" This is a book ";
        System.out.println(" The string s4 is :" +s4+"with length "+s4.length());
    }
}
```

```
s4=s4.trim();
System.out.println(" After trim() :" +s4+ " with length " +s4.length());
}
}
```

The output of above code will be

After trim() : Java with length 4

Length of string is 4

Because there is no space character in string

Ques.

Ans.

EX. NO: 3

DATE:

STRING BUFFER CLASS AND ITS METHODS

AIM:

To write a Java program using the methods of StringBuffer class.

ALGORITHM:

- Step 1: Initialize a string using StringBuffer constructor.
- Step 2: Calculate the length using length() method.
- Step 3: Print the character at the given position using charAt() method.
- Step 4: Replace a character using setCharAt() method.
- Step 5: Append a string at the end using append() method.
- Step 6: Insert a string in the middle using insert() method.
- Step 7: Delete a substring from the main string using delete() method.

PROGRAM:

```
class Str
{
    public static void main(String arg[])
    {
        StringBuffer sb=new StringBuffer("This is my college");
        System.out.println("This string sb is : " +sb);
        System.out.println("The length of the string sb is : " +sb.length());
        System.out.println("The character at an index of 6 is : " +sb.charAt(6));
        sb.setCharAt(3,'x');
        System.out.println("After setting char x at position 3 : " +sb);
        System.out.println("After appending : " +sb.append(" in arumbakkam "));
        System.out.println("After inserting : " +sb.insert(19,"DGVC "));
        System.out.println("After deleting : " +sb.delete(19,22));
    }
}
```

EX. NO: 4

DATE:

RANDOM CLASS

AIM:

To generate random numbers using java utility package.

ALGORITHM:

Step 1: Declare an object for random class.

Step 2: Initialize an array of size 10.

Step 3: Using nextInt() method, generate random numbers and assign to array a.

Step 4: Arrange the random values in ascending order and display maximum and minimum number in an array.

PROGRAM:

```
// Random number generation
import java.util.Random;
public class ArrayRandom
{
    public static void main(String args[])
    {
        Random r = new Random();
        int a[] = new int[10];
        for(int i = 0; i < 10; i++)
            a[i] = r.nextInt(50) + 1;
        System.out.println("Random Array generated is .....");
        for(int i = 0; i < 10; i++)
            System.out.println(a[i]);
        for( int i = 0; i < 10; i++ )
            for (int j=i+1;j<10;j++)
        {
            if(a[i] > a[j])
            {
                int t=a[i];
                a[i]=a[j];
                a[j]=t;
            }
        }
        System.out.println("The sorted array is.....");
        for (int i=0;i<10;i++)
            System.out.println(a[i]);
        System.out.println("Maximum Number in the above list is: "+ a[9]);
        System.out.println("Minimum Number in the above list is:" +a[0]);
    }
}
```

EX. NO: 7

DATE:

HIERARCHICAL INHERITANCE

AIM:

To illustrate the concept of hierarchical inheritance in Java.

ALGORITHM:

- Step 1: Design a class called "Player" with name as its member.
- Step 2: Write the constructor of the class Player to initialize the name.
- Step 3: Design 3 classes namely "Cricket", "Football", "Hockey" with Player as their super class.
- Step 4: Write a method in each class to print the name of the corresponding sports' player name.

PROGRAM:

```
// Program showing Hierarchical inheritance
class Player
{
    String name;
    Player(String nm)
    {
        name=nm;
    }
}

class Cricket extends player
{
    Cricket(String nm)
    {
        super(nm);
    }
    void play()
    {
        System.out.println("The name of the cricket player is "+name);
    }
}
```

```
class Football extends Player
{
    Football(String nm)
    {
        super(nm);
    }
    void play()
    {
        System.out.println("The name of the football player is "+name);
    }
}

class Hockey extends Player
{
    Hockey(String nm)
    {
        super(nm);
    }
    void play()
    {
        System.out.println("The name of the hockey player is "+name);
    }
}

public class MainPlayer
{
    public static void main(String args[])
    {
        cricket c=new cricket("Sachin Tendulkar");
        football f=new football("Pele");
        hockey h=new hockey("Helen Mary");
        c.play();
        f.play();
        h.play();
    }
}
```

EX. NO: 8
DATE:

MULTIPLE INHERITANCE USING INTERFACE

AIM:

To write a Java program for multiple inheritance using the concept of interface.

ALGORITHM:

- Step 1: Define an interface "Area" with a constant pi and abstract method calc().
- Step 2: Define a class "Shape" with a member radius.
- Step 3: Initialize radius using constructor Shape().
- Step 4: Define a sub-class "Circle" with two super classes- Area and Shape.
- Step 5: Compute the area and circumference of the circle and display the values.

PROGRAM:

```
interface Area
{
    final double pi=3.14;
    void calc();
}

class Shape
{
    double radius;
    Shape(double r)
    {
        radius=r;
    }
}

class Circle extends Shape implements Area
{
    Circle(double r)
    {
        super(r);
    }
    public void calc()
    {
        double area=pi*radius*radius;
        double circum=2.0*pi*radius;
        System.out.println ("Radius:"+radius);
        System.out.println ("Area of circle:"+area);
        System.out.println("Circumference of circle:"+circum);
    }
}
```

```
class Multiple
{
    public static void main(String args[])
    {
        Circle c=new Circle(15.4);
        c.calc();
    }
}
```

EX. NO: 9

DATE:

PACKAGES

AIM:

To write a Java program using packages.

ALGORITHM:

- Step 1: Define a package "pack" with methods performing basic arithmetic operations.
- Step 2: Save the package in the folder "pack" and compile it.
- Step 3: Write the main program importing the package "pack" and call the methods defined within the package.

PROGRAM:

```
import pack.*;  
import java.util.*;  
class MainPackage  
{  
    public static void main(String args[])  
    {  
        Scanner s=new Scanner(System.in);  
        int a,b;  
        System.out.println("Enter 2 numbers....");  
        a=Integer.parseInt(s.nextLine());  
        b=Integer.parseInt(s.nextLine());  
        Arithmetic obj = new Arithmetic();  
        System.out.println(a + " + " + b + " = " +obj.add(a,b));  
        System.out.println(a + " - " + b + " = " +obj.sub(a,b));  
        System.out.println(a + " * " + b + " = " +obj.mul(a,b));  
        System.out.println(a + " / " + b + " = " +obj.div(a,b));  
    }  
}  
  
package pack;  
public class Arithmetic  
{  
    public int add(int a,int b)  
    {  
        return(a+b);  
    }  
    public int sub(int c, int d)  
    {  
        return(c-d);  
    }  
}
```

```
public int mul(int e, int f)
{
    return(e*f);
}
public int div(int g, int h)
{
    return(g/h);
}
}
```

EX. NO: 10

DATE:

BUILT-IN EXCEPTION HANDLING

AIM:

To write a Java program to show the working of built-in exception handling.

ALGORITHM:

- Step 1: Divide an integer by 0 and raise "Divide by zero" exception.
- Step 2: Initialize the value of array greater than its size and raise "Array Index Out of Bounds" exception.
- Step 3: Instead of entering number, enter alphabetical characters and raise "Number Format" exception.
- Step 4: Define a class and methods for invoking above exceptions.

PROGRAM:

```
import java.util.*;
public class Exception
{
    void arith()
    {
        int a,b;
        try
        {
            System.out.println("1. Inside Arithmetic");
            a=0;
            b=10/a;
            System.out.println("a="+a+" b="+b);
        }
        catch(ArithmException e)
        {
            System.out.println("\t"+ "Divide by zero is an arithmetic
exception\n"+ "\t"+e);
        }
    }
    void array()
    {
        try
        {
            System.out.println("2. Inside Array");
            int a[]={1,2,3,4,5};
            a[10]=78;
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("\t"+ "Wrong index in the given
array"\n"+ "\t"+e);
        }
    }
}
```

```
void num()
{
    Scanner s=new Scanner(System.in);
    int x[]={};
    System.out.println("Enter 5 elements:");
    try
    {
        for (int i=0;i<5;i++)
            x[i]=Integer.parseInt(s.nextLine());
    }
    catch(NumberFormatException e)
    {
        System.out.println("\t"+"Wrong input"+ "\n" +"\t"+e);
    }
}
public static void main(String args[]) throws IOException
{
    Exception obj=new Exception();
    obj.arith();
    obj.array();
    obj.num();
}
```

EX. NO: 11

DATE:

USER-DEFINED EXCEPTION HANDLING

AIM:

To write a Java program for user-defined exception handling.

ALGORITHM:

- Step 1: Define a sub-class "MyException" with super class as "Throwable".
- Step 2: Declare arrays for account number, account holder names and balance maintained by them.
- Step 3: Define an user-defined exception "MyException" to raise exception for the account holders whose balance is less than 2000.

PROGRAM:

```
class MyException extends Throwable
{
int accno[] = {1001,1002,1003,1004,1005};
String name[] = {"Hari","Siva","Bhanu","Rama","Chandu"};
double bal[] = {2500,3500,1500,1000,6000};
    MyException(String S)
    {
        super(S);
    }
public static void main(String args[])
{
    MyException me = new MyException("");
    System.out.println("AccNo \t Name \t Balance ");
    for(int i=0;i<5;i++)
    {
        try
        {
            System.out.println(me.accno[i]+ "\t" + me.name[i] + "\t" +me.bal[i] );
            if( me.bal[i] < 2000 )
                throw new MyException("InsufficientBalance");
        }
        catch(MyException e)
        {
            System.out.println(e);
            continue;
        }
    }
}
```

EX. NO: 12

DATE:

THREADS

AIM:

To illustrate the concept of threads in Java.

ALGORITHM:

- Step 1: Create a class A which extends Thread. In the run() method, use the yield() method, to relinquish the control to other thread.
- Step 2: Create a class B which extends Thread. In the run() method, use the stop() method, to abort the thread.
- Step 3: Create a class C which extends Thread that runs normally.
- Step 4: Create objects for classes A,B,C and start them accordingly.

PROGRAM:

```
// Illustration about threads
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("Exited from A");
    }
}
class B extends Thread
{
    public void run()
    {
        for (int i=1;i<=5;i++)
        {
            if (i==3) stop();
            System.out.println("From thread B:i="+i);
        }
        System.out.println("Exited from B");
    }
}
```

```
class C extends Thread
{
    public void run()
    {
        for (int i=1;i<=5;i++)
            System.out.println("From thread C:i="+i);
        System.out.println("Exited from C");
    }
}
class Threads
{
    public static void main(String args[])
    {
        A a=new A();
        B b=new B();
        C c=new C();
        a.start();
        b.start();
        c.start();
    }
}
```

EX. NO: 13

DATE:

FILE MANIPULATION

AIM:

To copy the content of one file into another file using file streams.

ALGORITHM:

Step1: Read a file name from the user.

Step 2: Read a target file name from the user.

Step 3: Read the content of the source file name & copy the content as byte into target file name.

Step 4: Repeat step 3 until the source file reaches its end.

PROGRAM:

```
import java.io.*;
public class FileCopy
{
    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 existing 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 is copied");
        }
        catch(IOException e)
        {
            System.out.println(e);
        }
    }
}
```

EX. NO: 14

DATE:

WORKING WITH FONTS AND COLORS

AIM:

To execute an applet program in show the various fonts and colors.

ALGORITHM:

Step 1:Write the applet tag with code , height , width.

Step 2:In the paint method , do the following.

A:Set font f1,f2,f3,f4 with various font faces, styles & size.

B:Set color as gray , font as f1 & print string "Be happy & hopeful".

C:Set color as blue , font as f2 & print string " Be self confident".

D: Set color as green , font as f3 & print string "Be an optimistic Person".

E: Set color as red , font as f4 & print string "Always keep smiling".

Step 3: Run the applet using the command "appletviewer program.java".

PROGRAM:

```
// Changing fonts and colors
import java.awt.*;
import java.applet.*;
public class ColorApp extends Applet
{
    public void paint(Graphics g)
    {
        Font f1=new Font("TimesNewRoman", Font.BOLD,20);
        Font f2=new Font("Courier", Font.ITALIC,20);
        Font f3=new Font("Helvotica", Font.PLAIN,20);
        Font f4=new Font("TimesNewRoman", Font.BOLD,20);
        g.setColor(Color.gray);
        g.setFont(f1);
        g.drawString("Be happy and hopeful",30,30);
        g.setColor(Color.blue);
        g.setFont(f2);
        g.drawString("Be Self-Confident",30,70);
        g.setColor(Color.green);
        g.setFont(f3);
        g.drawString("Be an optimistic person",30,110);
        g.setColor(Color.red);
        g.setFont(f4);
        g.drawString("Always keep smiling",30,150);
    }
}
/*<applet code="ColorApp" height="30" width="300"></applet>*/
```

EX. NO: 15

DATE:

PARAMETER PASSING TECHNIQUE

AIM:

To write an applet program incorporating parameter passing technique.

ALGORITHM:

Step 1: In the applet tag, define two parameters.

Step 2: In the init() method, get those two parameters and compare them.

Step 3: In the paint() method, display whether two values are equal or which is greater and smaller.

PROGRAM:

```
/* <applet code="MyApplet2.class" width = 600 height= 450>
<param name = "t1" value="102">
<param name = "t2" value ="101">
</applet> */
import java.applet.*;
import java.awt.*;
public class MyApplet2 extends Applet
{
int n1,n2,n;
    public void init()
    {
        n1 = Integer.parseInt(getParameter("t1"));
        n2 = Integer.parseInt(getParameter("t2"));
    }
    public void paint(Graphics g)
    {
        g.drawString("First Value:"+String.valueOf(n1),100,50);
        g.drawString("Second Value:"+String.valueOf(n2),100,70);
        if(n1==n2)
            g.drawString("Both values are equal",100,90);
        else if(n1>n2)
            g.drawString(n1+ " is greater than "+n2,100,90);
        else
            g.drawString(n2+ " is greater than "+n1,100,90);
    }
}
```

EX. NO: 16

DATE:

DRAWING VARIOUS SHAPES

AIM:

To write a Java Applet program for incorporating graphics for drawing shapes.

ALGORITHM:

- Step 1: Write the applet tag with code , height , width.
- Step 2: In the paint method , draw shapes like line, rectangle , rounded rectangle , oval, circle, arc, polygon.
- Step 3: Set different colors for the shapes.
- Step 4: Run the applet using the command "appletviewer program.java".

PROGRAM:

```
// Drawing various shapes
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
/*<applet code="Shapes" height=1000 width=1000> </applet>*/
public class Shapes extends Applet
{
    public void init()
    {
        setForeground(Color.yellow);
        setBackground(Color.black);
    }
    public void paint(Graphics g)
    {
        g.drawString("DIFFERENT SHAPES",400,53);
        g.drawString("-----",400,60);
        g.drawString("1.LINE",125,90);
        g.drawLine(150,100,250,100);
        g.drawString("2.RECTANGLE",125,125);
        g.drawRect(150,150,70,40);
        g.drawString("3.ROUND RECTANGLE",125,220);
        g.drawRoundRect(150,250,70,40,10,20);
        g.drawString("4.CIRCLE",125,315);
        g.drawOval(150,340,50,50);
        g.drawString("5.ARC",125,420);
        g.drawArc(125,445,90,80,50,70);
    }
}
```

EX. NO: 17

DATE:

TEMPERATURE CONVERSION

AIM:

To write a Java applet program for converting farenheit to Celsius and viceversa using AWT controls like textbox, label, button etc.

ALGORITHM:

- Step 1: Write the applet tag with code=name of program, height & width of the browser.
- Step 2: Create 2 textboxes & label them accordingly.
- Step 3: Create 2 buttons, one for F2C (Fahrenheit to Celcius) & other for C2F (Celcius to Fahrenheit).
- Step 4: Write action listeners for 2 buttons using actionPerformed method.
- Step 5: If we press F2C, compute $F=5/9(C-32)$ & display 'f' as result.
- Step 6: If we press C2F, compute $C=9*F/5+32$ & display 'c' as result.

PROGRAM:

```
// Temperature conversion
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

//<applet code=Farcel width=500 height=500></applet>
public class Farcel extends Applet implements ActionListener
{
    Label l1,l2;
    TextField t1,t2;
    Button b1,b2;
    public void init()
    {
        l1=new Label("Input");
        l2=new Label("Output");
        b1=new Button("F2C");
        b2=new Button("C2F");
        t1=new TextField(5);
        t2=new TextField(5);
        add(l1);add(t1);
        add(l2);add(t2);
        add(b1);add(b2);
        b1.addActionListener(this);
        b2.addActionListener(this);
    }
}
```

```
public void actionPerformed(ActionEvent ae)
{
float f=0,i=0;
String s=ae.getActionCommand();
try
    {i=Float.parseFloat(t1.getText());}
catch(NumberFormatException e){}
if (s.equals("F2C"))
    f=5*(i-32)/9;
else
if (s.equals("C2F"))
    f=(9*i)/5+32 ;
t2.setText(Float.toString(f));
}
```