



PARALA MAHARAJA ENGINEERING COLLEGE, BERHAMPUR

Dept. of Computer Sc. & Engg.

**“OBJECT ORIENTED
PROGRAMMING USING
JAVA / JAVA
PROGRAMMING”
LABORATORY MANUAL
(3RD / 5TH SEMESTER CSE)**

Kalyan Kumar Jena

ASSIGNMENT NO. 1:

Write a program to display the addition result of any two integers.

ASSIGNMENT NO. 2:

Write a program for the swapping of two numbers.

ASSIGNMENT NO. 3:

Write a program to print all the prime numbers between 100 to 200.

ASSIGNMENT NO. 4:

Write a program to check whether a number is palindrome or not.

ASSIGNMENT NO. 5:

Write a program to display the student name, roll no and age of a student using class and object concept.

ASSIGNMENT NO. 6:

Write separate programs that shows the implementation of

- (i) static variable and static member function.**
- (ii) static block concept**

ASSIGNMENT NO. 7:

Write a program that shows the use of default, parameterized and copy constructor.

ASSIGNMENT NO. 8:

Write separate programs for method overloading and method overriding mechanism.

ASSIGNMENT NO. 9:

Write a program for single inheritance mechanism.

ASSIGNMENT NO. 10:

Write a program where abstract class concept is used.

ASSIGNMENT NO. 11:

Write a program to pass the arguments from child class constructor to parent class constructor using super ().

ASSIGNMENT NO. 12:

Write a program to display name, department and age of employee using interface mechanism.

ASSIGNMENT NO. 13:

Write a program for the implementation of user defined package.

ASSIGNMENT NO. 14:

Write a program for the manipulation of string using different functions of String class.

ASSIGNMENT NO. 15:

Write a program for the execution of multiple threads with their priority.

ASSIGNMENT NO. 16:

Write a complete program for the execution of applet mechanism.

ASSIGNMENT NO. 17:

Write a program for drawing several GUI components on applet.

ASSIGNMENT NO. 18:

Write a program for the implementation of several methods of Graphics class.

ASSIGNMENT NO. 19:

Write a program for handling multiple exceptions.

ASSIGNMENT NO. 20:

Write a complete program for Java Database Connectivity.

ASSIGNMENT NO. 21:

Write a program for the use of swing components.

ASSIGNMENT NO. 22:

Write a java socket programming in which client sends a text and server receives it.

ASSIGNMENT NO. 23:

Write a program for the multiplication of two matrix.

ASSIGNMENT NO. 24:

Write a program for the addition of two matrix.

ASSIGNMENT NO. 25:

Write a program to display the transpose of a matrix.

ASSIGNMENT NO. 1:

Write a program to display the addition result of any two integers.

PROGRAM:

```
class Add
{
public static void main(String args[ ])
{

int x = Integer.parseInt(args[0]);
int y = Integer.parseInt(args[1]);
int z=x+y;
System.out.println("The Sum="+z);
}
}
```

ASSIGNMENT NO. 2:

Write a program for the swapping of two numbers.

PROGRAM:

```
class Swap
{
public static void main(String args[ ])
{

int x = Integer.parseInt(args[0]);
int y = Integer.parseInt(args[1]);
x=x+y;
y=x-y;
x=x-y;
System.out.println("After swapping the two numbers are"+x+ " "+y);
}
}
```

ASSIGNMENT NO. 3:

Write a program to print all the prime numbers between 100 to 200.

PROGRAM:

```
class Prime
{
public static void main(String args[ ])
{
for(int i=100;i<=200;i++)
{
int c=0;
for(int j=1;j<=i;j++)
{
if(i%j==0)
{
c++;
}
}
if(c==2)
{
System.out.println(i);
}
}
}
}
```

ASSIGNMENT NO. 4:

Write a program to check whether a number is palindrome or not.

PROGRAM:

```
class Pali
{
public static void main(String args[ ])
{
int x = Integer.parseInt(args[0]);
int n,b,r=0;
n=x;
while(x>0)
{
b=x%10;
r=r*10+b;
x=x/10;
}
if(n==r)
{
System.out.println("The number is palindrome");
}
else
{
System.out.println("The number is not palindrome");
}
}
}
```

ASSIGNMENT NO. 5:

Write a program to display the student name, roll no and age of a student using class and object concept.

PROGRAM:

```
class Student
{
String n;
int r,a;
void accept(String i , int j , int k)
{
n=i;
r=j;
a=k;
}
void display( )
{
System.out.println("The Name="+n);
System.out.println("The Roll no="+r);
System.out.println("The Age="+a);
}
}
class Student1
{
public static void main(String args[ ])
{
String z=args[0];
int m=Integer.parseInt(args[1]);
int n=Integer.parseInt(args[2]);
```



```
Student k = new Student( );
```

```
k.accept(z,m,n);
```

```
k.display( );
```

```
}
```

```
}
```

ASSIGNMENT NO. 6:

Write separate programs that shows the implementation of

- (i) static variable and static member function.
- (ii) static block concept

PROGRAM:

(i)

```
class Use
{
int a;
static int b;
Use(int i,int j)
{
a=i;
b=j;
}
void show()
{
System.out.println(a);
}
static void display()
{
System.out.println(b);
}
}
class Use1
{
public static void main(String args[])
{
Use k=new Use(5,7);
k.show();
k.display();
}
}
```

(ii)

```
class Use
{
static int x;
static
{
x=10;
System.out.println(x);
}
public static void main(String args[])
{
System.out.println("hello");
}
}
```

ASSIGNMENT NO. 7:

Write a program that shows the use of default, parameterized and copy constructor.

PROGRAM:

```
class Use
{
int a,b;
Use() -----→Default Constructor
{
a=4;
b=9;
System.out.println(a+ " " +b);
}
Use(int i , int j) -----→ Parameterized Constructor
{
a=i;
b=j;
System.out.println(a+ " " + b);
}
Use(Use k) -----→Copy Constructor
{
a=k.a;
b=k.b;
System.out.println(a+ " " + b);
}
}
class Use1
{
public static void main( String args[])
{
Use k = new Use();
Use m = new Use(5,7);
```

Use z=new Use (m);

}

}

ASSIGNMENT NO. 8:

Write separate programs for method overloading and method overriding mechanism.

PROGRAM:

Method Overloading:

```
class Use
{
int a,b;
void display( int i)
{
a=i;
System.out.println(a);
}
void display(int i , int j)
{
a=i;
b=j;
System.out.println(a+ " " + b);
}
}
class Use1
{
public static void main(String args[] )
{
Use k = new Use();
k.display(6);
k.display(5,9);
}
}
```

Method Overriding:

```
class A
{
void show()
{
System.out.println("hello");
}
}
class B extends A
{
void show()
{
System.out.println("world");
}
}
class C
{
public static void main(String args[ ])
{
B k=new B();
A t=new A();
k.show();
t.show();
}
}
```

ASSIGNMENT NO. 9:

Write a program for single inheritance mechanism.

PROGRAM:

```
class A
{
void show( )
{
System.out.println("hello");
}
}
class B extends A
{
void disp( )
{
System.out.println("world");
}
}
class C
{
public static void main(String args[ ])
{
B k=new B( );
k.show( );
k.disp( );
}
}
```


ASSIGNMENT NO. 10:

Write a program where abstract class concept is used.

PROGRAM:

```
abstract class A
{
void show();

}
class B extends A
{
void show()

{
System.out.println("world");
}
}
class C
{
public static void main(String args[ ])
{
B k=new B();
k.show();
}
}
```

ASSIGNMENT NO. 11:

Write a program to pass the arguments from child class constructor to parent class constructor using super ().

PROGRAM:

```
class A
{
int x,y;
A( int i , int j)
{
x=i;
y=j;
}
}
class B extends A
{
int z;

B(int i,int j,int k )

{
super(i,j);
z=k;
}
void disp( )
{
System.out.println(x+y+z);
}
}
class C
{
public static void main(String args[ ])
{
B k=new B(10,20,30 );
k.disp( );
}
}
```

ASSIGNMENT NO. 12:

Write a program to display name, department and age of employee using interface mechanism.

PROGRAM:

```
public interface A
{
void name(String n);
void dept(String d);
void age(int a);
}

public class B implements A
{
void name(String n)
{
System.out.println(n);
}

void dept(String d)
{
System.out.println(d);
}

void age(int a)
{
System.out.println(a);
}

public static void main(String args[ ])
{
B k=new B( );
k.name("Ram");
k.dept("finance");
}
```

```
k.age(30);
```

```
}
```

```
}
```

ASSIGNMENT NO. 13:

Write a program for the implementation of user defined package.

PROGRAM:

File-1

```
package Color;

public class Red

{

public void disp1()

{

System.out.println("Red");

}

}
```

File-2

```
package Color;

public class Green

{

public void disp2()

{

System.out.println("Green");

}

}
```

File-3

```
package Color;

public class Blue

{

public void disp3()

{

System.out.println("Blue");

}
```

```
}
```

Implementation File

```
import Color.*;
```

```
class Pack
```

```
{
```

```
Public static void main(String args[ ])
```

```
{
```

```
Red r=new Red( );
```

```
Green g=new Green( );
```

```
Blue b=new Blue( );
```

```
r.disp1( );
```

```
g.disp2( );
```

```
b.disp3( );
```

```
}
```

```
}
```

ASSIGNMENT NO. 14:

Write a program for the manipulation of string using different functions of String class.

PROGRAM:

```
class Str
{
public static void main(String args[ ])
{
String s1=new String("Hello World");
System.out.println(s1.indexOf('l'));
System.out.println(s1.lastIndexOf('l'));
System.out.println(s1.substring(4));
String s2=s1.toUpperCase( );
String s3=s1.toLowerCase( );
System.out.println(s1+ " " +s2+ " " +s3);
}
}
```

ASSIGNMENT NO. 15:

Write a program for the execution of multiple threads with their priority.

PROGRAM:

```
class Use1 extends Thread
{
public void run()
{
System.out.println("Thread1");
}
}

class Use2 extends Thread
{
public void run()
{
System.out.println("Thread2");
}
}

class Use3 extends Thread
{
public void run()
{
System.out.println("Thread3");
}
}

class Use
{
public static void main(String args[ ])
{
```



```
Use1 t1=new Use1();  
Use2 t2=new Use2();  
Use3 t3=new Use3();  
t1.setPriority(Thread.MAX_PRIORITY);  
t2.setPriority(6);  
t3.setPriority(t2.getPriority()+1);  
  
t1.start();  
t2.start();  
t3.start();  
  
}  
}
```

ASSIGNMENT NO. 16:

Write a complete program for the execution of applet mechanism.

PROGRAM:

```
import java.awt.*;
import java.applet.*;
public class Alife extends Applet
{
public void init()
{
System.out.println("initialized");
}
public void start()
{
System.out.println("started");
}
public void paint(Graphics g)
{
System.out.println("painting");
g.drawString("My Applet",10,100);
}
public void stop()
{
System.out.println("stopping");
}
public void destroy( )
{
System.out.println("destroying");
}
}
```

HTML File:

```
<html>
```

```
<body>
```

```
<applet code="Alife.class" width=300 height=300>
```

```
</applet>
```

```
</body>
```

```
</html>
```

ASSIGNMENT NO. 17:

Write a program for drawing several GUI components on applet.

PROGRAM:

```
import java.awt.*;
import java.applet.*;
public class Brder extends Applet
{
public void init()
{
setLayout(new BorderLayout( ));
setFont(new Font("SansSerif",Font.BOLD,32));
Button b1=new Button ("ok");
Button b2=new Button ("cancel");
Button b3=new Button ("abort");
Button b4=new Button ("retry");
TextArea t1=new TextArea(5,5);
add("East",b1);
add("West",b2);
add("North",b3);
add("South",b4);
add("Center",t1);
}
}
```

ASSIGNMENT NO. 18:

Write a program for the implementation of several methods of Graphics class.

PROGRAM:

```
import java.awt.*;
import java.applet.*;
public class Use extends Applet
{
public void paint(Graphics g)
{
g.drawLine(20,20,90,90);
g.drawRect(20,70,90,90);
g.drawOval(20,20,200,120);
g.setColor(Color.Green);
g.fillOval(70,30,100,100);
g.drawRoundRect(10,100,80,50,10,10);
g.fillRoundRect(20,110,60,30,5,5);
}
}
```

ASSIGNMENT NO. 19:

Write a program for handling multiple exceptions.

PROGRAM:

```
class Exc
{
public static void main(String args[ ])
{
try
{
int x=Integer.parseInt(args[0]);
int y=Integer.parseInt(args[1]);
int z=x/y;
System.out.println("result="+z);
}
catch (ArithmeticException ac)
{
System.out.println("Denominator must be non-zero");
}
catch (NumberFormatException n)
{
System.out.println("provide integer data");
}
catch (ArrayIndexOutOfBoundsException a)
{
System.out.println("give arguments");
}}}

```

ASSIGNMENT NO. 20:

Write a complete program for Java Database Connectivity.

PROGRAM:

```
import java.sql.*;

import java.io.*;

class Conn

{

public static void main(String args[])

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:DSN","scott","tiger");

Statement st=con.createStatement( );

ResultSet rs=st.executeQuery("select * from Emp");

While(rs.next( ))

{

System.out.println(rs.getInt("empno")+":"+rs.getString("ename"));

}

Con.close();

}
```

ASSIGNMENT NO. 21:

Write a program for the use of swing components.

PROGRAM:

```
import javax.swing.*;
public class FirstSwingExample {
public static void main(String[] args) {
JFrame f=new JFrame();//creating instance of JFrame

JButton b=new JButton("click");//creating instance of JButton
b.setBounds(130,100,100, 40);//x axis, y axis, width, height

f.add(b);//adding button in JFrame

f.setSize(400,500);//400 width and 500 height
f.setLayout(null);//using no layout managers
f.setVisible(true);//making the frame visible
}
}
```


ASSIGNMENT NO. 22:

Write a java socket programming in which client sends a text and server receives it.

PROGRAM:

File: MyServer.java

```
import java.io.*;
import java.net.*;
public class MyServer {
public static void main(String[] args){
try{
ServerSocket ss=new ServerSocket(6666);
Socket s=ss.accept();//establishes connection
DataInputStream dis=new DataInputStream(s.getInputStream());
String str=(String)dis.readUTF();
System.out.println("message= "+str);
ss.close();
}catch(Exception e){System.out.println(e);}
}
}
```

File: MyClient.java

```
import java.io.*;
import java.net.*;
public class MyClient {
public static void main(String[] args) {
try{
Socket s=new Socket("localhost",6666);
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
dout.writeUTF("Hello Server");
dout.flush();
dout.close();
s.close();
}catch(Exception e){System.out.println(e);}
}
}
```

ASSIGNMENT NO. 23:

Write a program for the multiplication of two matrix.

PROGRAM:

```
import java.util.Scanner;

class MatrixMultiplication
{
    public static void main(String args[])
    {
        int m, n, p, q, sum = 0, c, d, k;

        Scanner in = new Scanner(System.in);
        System.out.println("Enter the number of rows and columns of first matrix");
        m = in.nextInt();
        n = in.nextInt();

        int first[][] = new int[m][n];

        System.out.println("Enter the elements of first matrix");

        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                first[c][d] = in.nextInt();

        System.out.println("Enter the number of rows and columns of second matrix");
        p = in.nextInt();
        q = in.nextInt();

        if ( n != p )
            System.out.println("Matrices with entered orders can't be multiplied with each other.");
        else
        {
            int second[][] = new int[p][q];
            int multiply[][] = new int[m][q];

            System.out.println("Enter the elements of second matrix");

            for ( c = 0 ; c < p ; c++ )
                for ( d = 0 ; d < q ; d++ )
                    second[c][d] = in.nextInt();

            for ( c = 0 ; c < m ; c++ )
            {
                for ( d = 0 ; d < q ; d++ )
                {
                    for ( k = 0 ; k < p ; k++ )
                    {
                        sum = sum + first[c][k]*second[k][d];
                    }

                    multiply[c][d] = sum;
                    sum = 0;
                }
            }
        }
    }
}
```

```
}  
  
    System.out.println("Product of entered matrices:-");  
  
for ( c = 0 ; c < m ; c++ )  
{  
    for ( d = 0 ; d < q ; d++ )  
        System.out.print(multiply[c][d]+" ");  
  
    System.out.print("\n");  
}  
}  
}
```

ASSIGNMENT NO. 24:

Write a program for the addition of two matrix.

PROGRAM:

```
import java.util.Scanner;

class AddTwoMatrix
{
    public static void main(String args[])
    {
        int m, n, c, d;
        Scanner in = new Scanner(System.in);

        System.out.println("Enter the number of rows and columns of matrix");
        m = in.nextInt();
        n = in.nextInt();

        int first[][] = new int[m][n];
        int second[][] = new int[m][n];
        int sum[][] = new int[m][n];

        System.out.println("Enter the elements of first matrix");

        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                first[c][d] = in.nextInt();

        System.out.println("Enter the elements of second matrix");

        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                second[c][d] = in.nextInt();

        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                sum[c][d] = first[c][d] + second[c][d]; //replace '+' with '-' to subtract matrices

        System.out.println("Sum of entered matrices:-");

        for ( c = 0 ; c < m ; c++ )
        {
            for ( d = 0 ; d < n ; d++ )
                System.out.print(sum[c][d]+" ");

            System.out.println();
        }
    }
}
```

ASSIGNMENT NO. 25:

Write a program to display the transpose of a matrix.

PROGRAM:

```
import java.util.Scanner;

class TransposeAMatrix
{
    public static void main(String args[])
    {
        int m, n, c, d;

        Scanner in = new Scanner(System.in);
        System.out.println("Enter the number of rows and columns of matrix");
        m = in.nextInt();
        n = in.nextInt();

        int matrix[][] = new int[m][n];

        System.out.println("Enter the elements of matrix");

        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                matrix[c][d] = in.nextInt();

        int transpose[][] = new int[n][m];

        for ( c = 0 ; c < m ; c++ )
        {
            for ( d = 0 ; d < n ; d++ )
                transpose[d][c] = matrix[c][d];
        }

        System.out.println("Transpose of entered matrix:-");

        for ( c = 0 ; c < n ; c++ )
        {
            for ( d = 0 ; d < m ; d++ )
                System.out.print(transpose[c][d]+" ");

            System.out.print("\n");
        }
    }
}
```