

## WEEK - 9

Week-9

a. Write a java program that takes tab separated data (one record per line) from a text file and insert them into a database.

Aim: To Write a java program that takes tab separated data (one record per line) from a text file and insert them into a database.

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

// Class
class GFG {

    // Main driver method
    public static void main(String[] args)
    {

        // The file reading process may sometimes give
        // IOException

        // Try block to check for exceptions
        try {

            // Creating a FileReader object and
            // file to be read is passed as in parameters
            // from the local directory of computer
            FileReader fr = new FileReader("gfgInput.txt");

            // FileReader will open that file from that
            // directory, if there is no file found it will
            // through an IOException

            // Creating a FileWriter object
            FileWriter fw = new FileWriter("gfgOutput.txt");

            // It will create a new file with name
            // "gfgOutput.txt", if it is already available,
            // then it will open that instead

            // Declaring a blank string in which
            // whole content of file is to be stored
            String str = "";

            int i;
```

```

// read() method will read the file character by
// character and print it until it end the end
// of the file

// Condition check
// Reading the file using read() method which
// returns -1 at EOF while reading
while ((i = fr.read()) != -1) {

    // Storing every character in the string
    str += (char)i;
}

// Print and display the string that
// contains file data
System.out.println(str);

// Writing above string data to
// FileWriter object
fw.write(str);

// Closing the file using close() method
// of Reader class which closes the stream &
// release resources that were busy in stream
fr.close();
fw.close();

// Display message
System.out.println(
    "File reading and writing both done");
}

// Catch block to handle the exception
catch (IOException e) {

    // If there is no file in specified path or
    // any other error occurred during runtime
    // then it will print IOException

    // Display message
    System.out.println(
        "There are some IOException");
}
}

```

B. Write a java program that prints the metadata of a given table.

AIM:TO Write a java program that prints the metadata of a given table.

```
import java.sql.*;
class Dbmd{
public static void main(String args[]){
try{
Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
DatabaseMetaData dbmd=con.getMetaData();

System.out.println("Driver Name: "+dbmd.getDriverName());
System.out.println("Driver Version: "+dbmd.getDriverVersion());
System.out.println("UserName: "+dbmd.getUserName());
System.out.println("Database Product Name: "+dbmd.getDatabaseProductName());
System.out.println("Database Product Version: "+dbmd.getDatabaseProductVersion());

con.close();
}catch(Exception e){ System.out.println(e);}
}
}
```

ADVERTISEMENT

```
Output:Driver Name: Oracle JDBC Driver
      Driver Version: 10.2.0.1.0XE
      Database Product Name: Oracle
      Database Product Version: Oracle Database 10g Express Edition
                               Release 10.2.0.1.0 -Production
```

download this example

ADVERTISEMENT

ADVERTISEMENT

Example of DatabaseMetaData interface that prints total number of tables :

```
import java.sql.*;
class Dbmd2{
public static void main(String args[]){
try{
Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");

DatabaseMetaData dbmd=con.getMetaData();
String table[]{"TABLE"};
ResultSet rs=dbmd.getTables(null,null,null,table);

while(rs.next()){
System.out.println(rs.getString(3));
}
}
```

```

con.close();

}catch(Exception e){ System.out.println(e);}

}
}
ADVERTISEMENT
ADVERTISEMENT
download this example
ADVERTISEMENT
ADVERTISEMENT
Example of DatabaseMetaData interface that prints total number of views :
import java.sql.*;
class Dbmd3{
public static void main(String args[]){
try{
Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");

DatabaseMetaData dbmd=con.getMetaData();
String table[]{"VIEW"};
ResultSet rs=dbmd.getTables(null,null,null,table);

while(rs.next()){
System.out.println(rs.getString(3));
}

con.close();

}catch(Exception e){ System.out.println(e);}

}
}

```

Output:

```

Driver Name: Oracle JDBC Driver
Driver Version: 10.2.0.1.0XE
Database Product Name: Oracle
Database Product Version: Oracle Database 10g Express Edition
Release 10.2.0.1.0 -Production

```

Week-10

A. Write a java program that simulates a traffic light. The program lets the user select one of

three

lights: Red, Yellow or Green with radio buttons. On selecting a button an appropriate message with "STOP" or "READY" or "GO" should appear above the buttons in selected color. Initially, there is no message shown.

Aim: To write a java program that simulates a traffic light. The program lets the user select one of

three lights: red, yellow or green. When a radio button is selected, the light is turned on and only one light can be on at a time. No light is on when the program starts.

Program:

```
import java.awt.*;
import java.awt.event.*;
class MyFrame extends Frame implements ItemListener
{
    Checkbox b1, b2, b3;
    CheckboxGroup cbg;
    MyFrame()
    { setLayout( new FlowLayout() );
      cbg = new CheckboxGroup();
      b1 = new Checkbox("RED",cbg,false);
      b2 = new Checkbox("GREEN",cbg,false);
      b3 = new Checkbox("YELLOW",cbg,false);
      add(b1); add(b2); add(b3);
      b1.addItemListener(this);
      b2.addItemListener(this);
      b3.addItemListener(this);
      addWindowListener( new WindowAdapter()
      {
          public void windowClosing(WindowEvent we)
          {
              System.exit(0);
          }
      });
    }
    public void itemStateChanged(ItemEvent ie)
    { repaint();
    }
    public void paint(Graphics g)
    {
        if(b1.getState() )
        { g.setColor(Color.red);
          g.fillOval(100,100,100,100);
        }
        if(b2.getState())
        { g.setColor(Color.green);
          g.fillOval(100,100,100,100);
        }
        if(b3.getState())
```

```

{ g.setColor(Color.yellow);
g.fillOval(100,100,100,100);
}
}
}
class TLight
{
public static void main(String args[])
{MyFrame ob = new MyFrame();
ob.setTitle("Traffic Light");
ob.setSize(600, 300);
ob.setVisible(true);
}
}

```

Week-11

a. Write a java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. Use adapter classes.

Aim: To write a java program for handling mouse events.

Program:

```

import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class Mouse extends Applet implements MouseListener, MouseMotionListener
{
String txt="";
int x=10,y=30;
public void init()
{
addMouseListener(this);
addMouseMotionListener(this);
}
public void mouseClicked(MouseEvent me)
{
txt="Mouse Clicked";
setForeground(Color.pink);
repaint();
}
public void mouseEntered(MouseEvent me)
{
txt="Mouse Entered";
repaint();
}
public void mouseExited(MouseEvent me)
{

```

```

txt="Mouse Exited";
setForeground(Color.blue);
repaint();
}
public void mousePressed(MouseEvent me)
{
txt="Mouse Pressed";
setForeground(Color.blue);
repaint();
}
public void mouseMoved(MouseEvent me)
{
txt="Mouse Moved";
setForeground(Color.red);
repaint();
}
public void mouseDragged(MouseEvent me)
{
txt="Mouse Dragged";
setForeground(Color.green);
repaint();
}
public void mouseReleased(MouseEvent me)
{
txt="Mouse Released";
setForeground(Color.yellow);
repaint();
}
public void paint(Graphics g)
{
g.drawString(txt,30,40);
showStatus("Mouse events Handling");
}
}

```

B. Write a java program to demonstrate the key event handlers.

Aim: To Write a Java program for handling Key events.

Program:

```

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class Key extends Applet implements KeyListener
{
int x=20,y=30;
String msg="KeyEvents--->";
public void init()
{

```

```

addKeyListener(this);
requestFocus();
setBackground(Color.green);
setForeground(Color.blue);
}
public void keyPressed(KeyEvent k)
{
showStatus("KeyDown");
int key=k.getKeyCode();
switch(key)
{
case KeyEvent.VK_UP:
showStatus("Move to Up");
break;
case KeyEvent.VK_DOWN:
showStatus("Move to Down");
break;
case KeyEvent.VK_LEFT:
showStatus("Move to Left");
break;
case KeyEvent.VK_RIGHT:
showStatus("Move to Right");
break;
}
repaint();
}
public void keyReleased(KeyEvent k)
{
showStatus("Key Up");
}
public void keyTyped(KeyEvent k)
{
msg+=k.getKeyChar();
repaint();
}
public void paint(Graphics g)
{
g.drawString(msg,x,y);
}
}

```

WEEK -12

A. Write a java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, \*, % operations. Add a text field to display the result. Handle any possible exception like divided by zero.



AIM:Aim:To write a java program that works as a simple calculator. Use a grid layout to arrange buttons

for the digits +,-,\*,/,% operations. Add a text field to display the result.

Program:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class Calculator extends JFrame implements ActionListener
{
    Container c;
    JTextField t1;
    String a="",b;
    String oper="",s="",p="";
    int first=0,second=0,result=0;
    JButton b0,b1,b2,b3,b4,b5,b6,b7,b8,b9;
    JButton add,sub,mul,div,mod,res,clear;
    JPanel p1,p2;
    Calculator()
    {
        c =getContentPane();
        p1 = new JPanel();
        p2 = new JPanel(new GridLayout(4,4));
        t1=new JTextField(a,10);
        c.setLayout(new GridLayout(2,1));
        b0=new JButton("0");
        b1=new JButton("1");
        b2=new JButton("2");
        b3=new JButton("3");
        b4=new JButton("4");
        b5=new JButton("5");
        b6=new JButton("6");
        b7=new JButton("7");
        b8=new JButton("8");
        b9=new JButton("9");
        add=new JButton("+");
        sub=new JButton("-");
        mul=new JButton("*");
        div=new JButton("/");
        mod=new JButton("%");
        res=new JButton("=");
        clear = new JButton("CE");
        b0.addActionListener(this);
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
        b6.addActionListener(this);
```

```

b7.addActionListener(this);
b8.addActionListener(this);
b9.addActionListener(this);
add.addActionListener(this);
sub.addActionListener(this);
mul.addActionListener(this);
div.addActionListener(this);
mod.addActionListener(this);
res.addActionListener(this);
clear.addActionListener(this);
p1.add(t1); p1.add(clear);
p2.add(b0); p2.add(b1); p2.add(b2);
p2.add(b3); p2.add(b4); p2.add(b5);
p2.add(b6); p2.add(b7); p2.add(b8);
p2.add(b9); p2.add(add); p2.add(sub);
p2.add(mul); p2.add(div); p2.add(mod);
p2.add(res);
c.add(p1);
c.add(p2);
}
public void actionPerformed(ActionEvent ae)
{
a=ae.getActionCommand();
if(a=="0" || a=="1" || a=="2" || a=="3" || a=="4" || a=="5"
|| a=="6" || a=="7" || a=="8" || a=="9")
{
t1.setText(t1.getText()+a);
}
if(a=="+" || a=="-" || a=="*" || a=="/" || a=="%")
{
first = Integer.parseInt(t1.getText());
oper = a;
t1.setText("");
}
if(a=="=")
{
if(oper=="+")
result=first+Integer.parseInt(t1.getText());
if(oper=="-")
result=first-Integer.parseInt(t1.getText());
if(oper=="*")
result=first*Integer.parseInt(t1.getText());
if(oper=="/")
result=first/Integer.parseInt(t1.getText());
if(oper=="%")
result=first%Integer.parseInt(t1.getText());
t1.setText(result+"");
}
}

```

```

if(a == "CE")
t1.setText("");
}
public static void main(String args[])
{
Calculator ob = new Calculator();
ob.setSize(200,200);
ob.setVisible(true);
ob.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}

```

Week-13

a. Develop an applet that displays a simple message.

Aim: To develop an applet that displays a simple message

Program:

```

import java.io.*;
import java.awt.*;
import java.applet.Applet;
public class App1 extends Applet
{
public void paint(Graphics g)
{
g.drawString("MCA III semester Students",50,60);
setForeground(Color.blue);
}
}

```

B. Develop an applet that receives an integer in one text field and computes its factorial value and returns it in another text field, when the button named "compute" is clicked.

Aim: To develop an applet that receives an integer in one text field, and computes as factorial value and returns it in another text field, when the button named "Compute" is clicked

Program:

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.applet.Applet;
public class App2 extends Applet implements ActionListener
{
JTextField tf1;
JTextField tf2;
JButton b;
JLabel l;
public void init()
{

```

```
l=new JLabel("Enter the number & press the button");
tf1=new JTextField("",5);
tf2=new JTextField("",10);
b=new JButton("Compute");
add(l);
add(tf1);
add(tf2);
add(b);
b.addActionListener(this);
}
public void actionPerformed(ActionEvent ae)
{
String str=ae.getActionCommand();
String str1;
int fact=1;
if(str=="Compute")
{
int n=Integer.parseInt(tf1.getText());
for(int i=1;i<=n;i++)
fact=fact*i;
str1="" +fact;
tf2.setText(str1);
}
}
}
```