# Government Polytechnic College Aurangabad

**(**An Autonomous Institute of Technical Education)

"PERSUIT FOR EXCELLENCE"

# JAVA PROJECT

**Java Editor**

Submitted By-

Vaishnavi Nighvekar (177043)

Ganesh Pandit (177045)

Aniket Yadav (177063)

Guided by-

Ingle Madam

DEPARTMENT OF INFORMATION TECHNOLOGY

ACADEMIC YEAR 2019-2020

## DEPARTMENT OF INFORMATION TECHNOLOGY

**GOVERNMENT POLYTECHNIC, AURANGABAD**

**(An Autonomous Institute of Government of Maharashtra)**

# CERTIFICATE

This is to certify that Miss**. VAISHNAVI NIGHVEKAR, GANESH PANDIT and ANIKET YADAV** has successfully completed java mini project titled "**Java Editor"** during the academic year 2019-2020, in partial fulfilment of Diploma in **Information Technology** of Government Polytechnic, Aurangabad. To the best of my knowledge and belief this seminar work has not been submitted elsewhere.

**Date : 18/5/2020**

**Prof. R. Ingle**
**Project Guide**

**Prof. M. A. Dhaygude**
**H.O.D, I. T**

**Prof. F. A. Khan**
**PRINCIPAL**

# ABSTRACT

There are number of coding editors available in order to code in particular language and one of the most used programming languages is Java. So every time to use java, mostly we need to download and install another editor or IDE for that language. This seems tedious process and storage wasting of system. So we proposed an editor named "Java Editor" which will offer easy access to java language and will support many type of other functions and its very friendly to use and understand. The software specifically designed for Educational institutions where PC's seems to below ended that is having low RAM and disk space.

User will be able to create new file and saved with particular extension, and the program will run according to extension on single click. User will be able to import packages, methods and other basic syntax within the specific language just by clicking it from side panel. This Editor will be very helpful to write a particular program without any hesitation and will also be very helpful to every java programmer, because it will offer “one click run”, “one click copy” functions.

# INTRODUCTION

The **Java Editor** is a type of very simple editor specially designed for java and is used in java console programing and normal java programs. The screen for java editor is divided into two parts 1) The plain TextArea where you can easily write programs and can also adjust its size by expanding the window size, the default font size of this text area is 10 and default font style is calibre. 2) The second part of the screes contains the tree, where all the regular functions, class, package, and other keyword are defined. You can easily just double click on the tree to copy the corresponding name of the tree to the text area.

All this type of functions of the editor will help the user to type the program conveniently and efficiently. Rather than this functioning there is a status bar and a menubar in the editor window which will help you in other options.

- Status Bar-

  The status bar is a very important component of the window, it helps the user to understand the cursor's position and the size of the window. When the user will resize the window the size in the bottom right corner will automatically change and it will show you the accurate size of the window with its height and width. At the left hand side there is a line and column label, this label will help to see the cursors current position. The label will change continuously on any keep pressed and then it will show you the accurate position of the cursor.
- Menu Bar-

  This is a most important element of the editor window as it contains every functioning of the editor. The editor will help you and guide you to save, open, close, and to create new file for the editor. There is also a menu for Editing which contains the options such as cut, copy, paste and print the file. The next menu is format menu and this menu will help you so that you can adjust the font style and font size for your file. The last menu is Run which have two sub menu, they are **Run Java** and **Run appletviewer**. As there name suggest what this options actually do is to run the java program and the applet program separately.

# FEATURES

**The Java Editor contains the following features:**

1. **Friendly GUI :-**

   Rather than the other Editor that are very hard to understand and a bit of confusing, the java Editor is very easy to learn and it's also very easy to use for a fresh user.

2. **Easy Working:-**

   This Editor will help you to run the program easily and only in one click other changes in Editor are also very easy to implement.

3. **Click & Copy:-**

   The click and copy function help the user to copy the multiple statements program in just 2 clicks. This will help the user to type the program without any efforts.

4. **Useful Tools:-**

   The java Editor contains the status bar which will help the user to keep the track of his/her program, it is done by showing the line and columns number at the bottom.

5. **Tree View:-**

   The Tree view will help the user to differentiate between the various function, packages and other conditional statements and between many keyword.

# ADVANTAGES AND DISADVANTAGES

**Advantages:-**

- Provides “one click run” and “one click copy” functions.
- Provides easy way to run and type the program.
- Provides Format support with different font styles and sizes.
- Allow to use applet in your program.
- Tree structure will help to differentiate every class, packages, and many functions.

**Disadvantages:-**

- Unable to perform the programs for servlet and jsp.
- The limitation for the programs is only up to applet, AWT and swing.
- Lack to different frameworks such as “struds” and “spring”
- Unable to identify the shortcuts for the save and open.

# IMPLIMENTATION

This phase will content the screen shorts for our projects and its working.

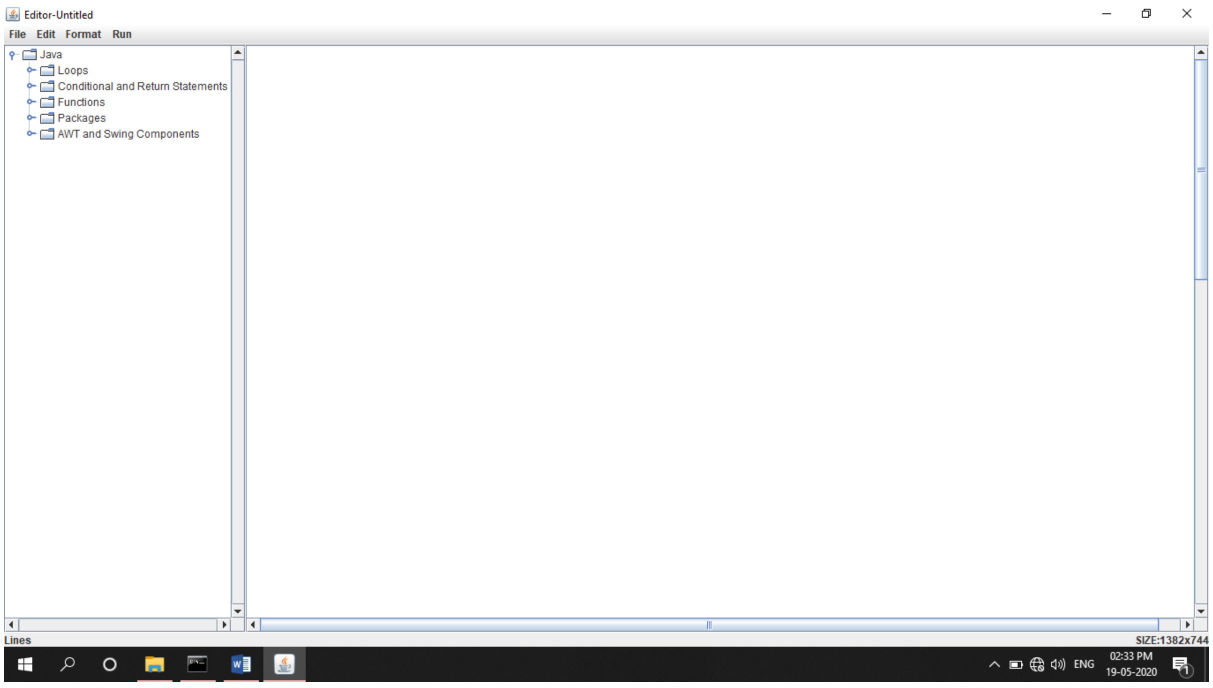


*Figure 1: Start Page*

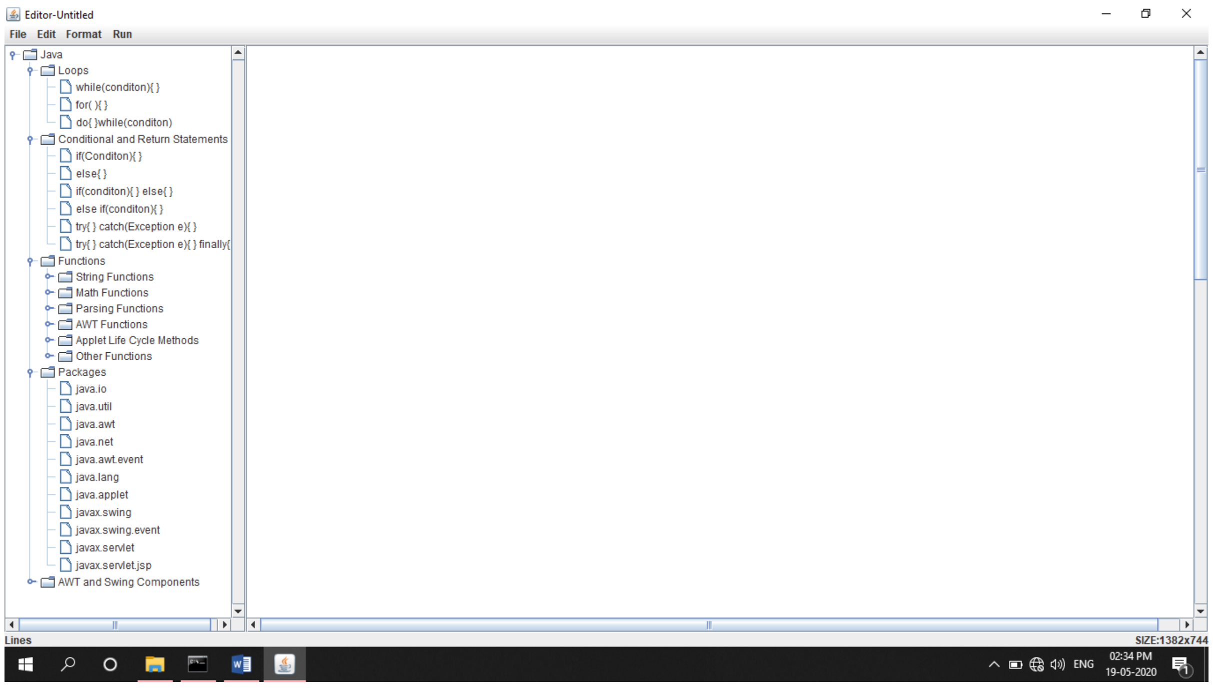


*Figure 2: Tree Structure*

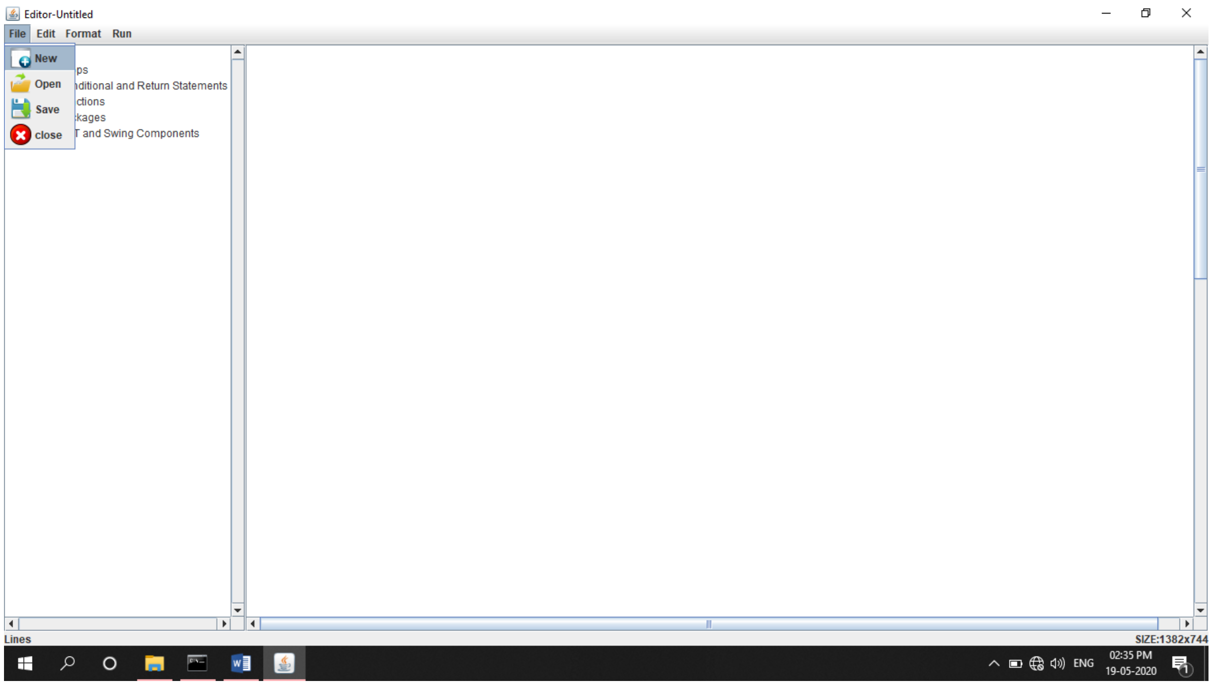


*Figure 3: File Menu*

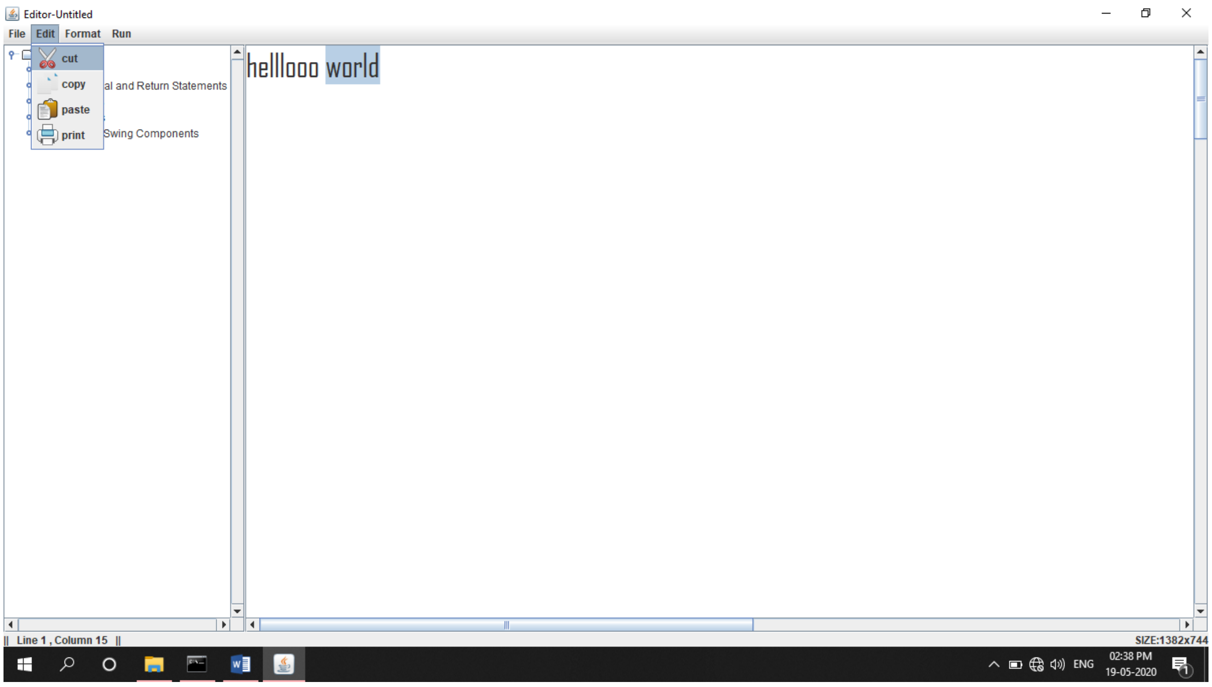
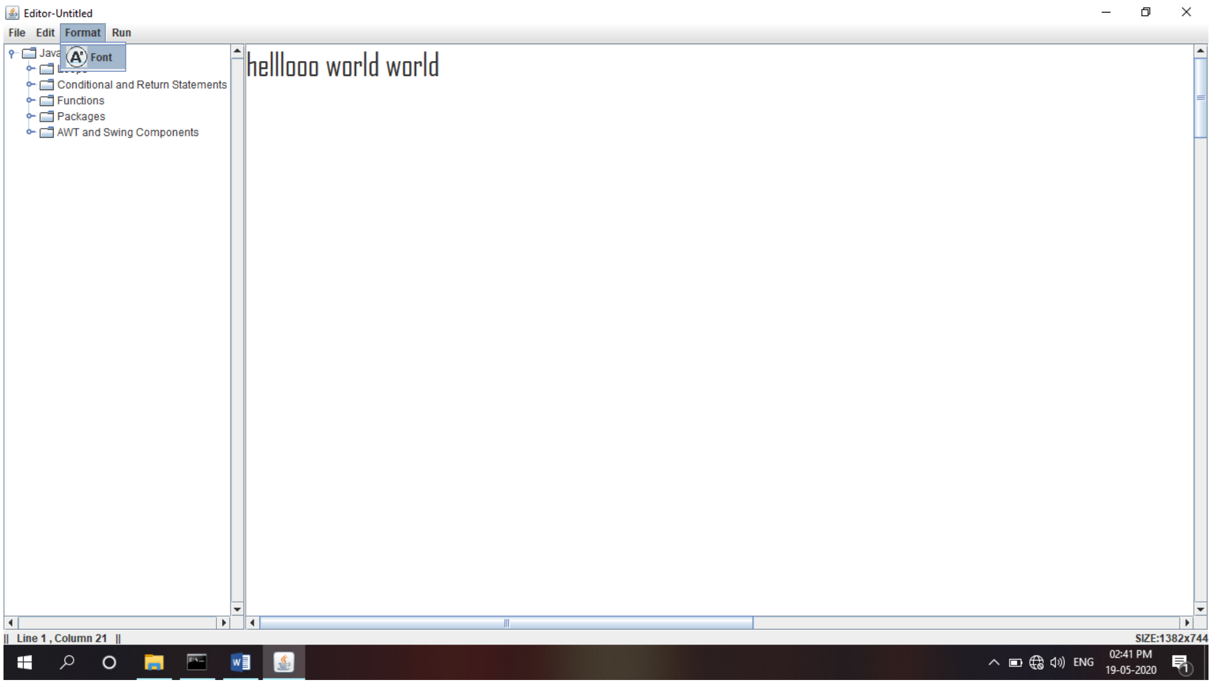
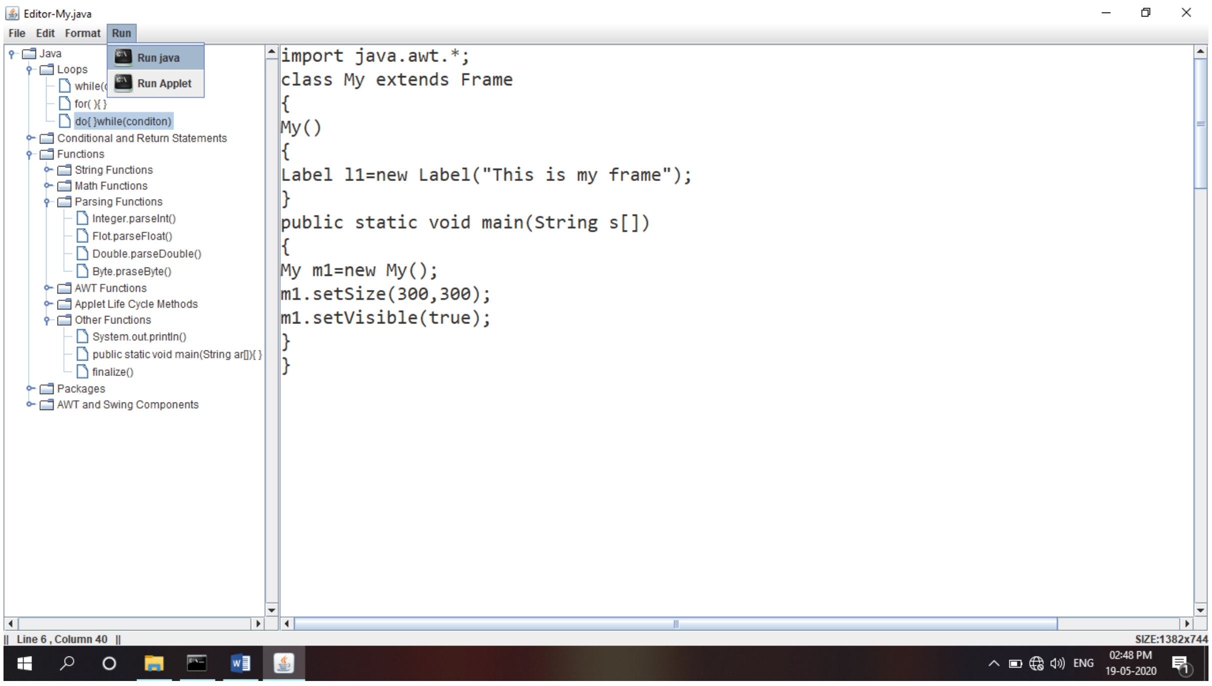


*Figure 4: Edit Menu*



```
helllooo world world
```

*Figure 5: Format Menu*



```
import java.awt.*;
class My extends Frame
{
My()
{
Label l1=new Label("This is my frame");
}
public static void main(String s[])
{
My m1=new My();
m1.setSize(300,300);
m1.setVisible(true);
}
}
```

*Figure 6: Run Menu*

# CODE

1) **Editor.java:-**

```
import java.awt.*;

import javax.swing.*;

import javax.swing.event.*;

import java.io.*;

import java.awt.event.*;

import javax.swing.plaf.metal.*;

import javax.swing.text.*;

import javax.swing.tree.*;

class Editor extends JFrame implements ActionListener {

    JTextArea t;

    JFrame f;

    String name,path;

    Font m_font;

    String nodename;

    Editor()

    {

      nodename="";

        f = new JFrame("Editor-Untitled");

     f.setLayout(new BorderLayout());

     f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        try {

            // Set metl look and feel
UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndF
eel");


            // Set theme to ocean

            MetalLookAndFeel.setCurrentTheme(new
OceanTheme());

        }

        catch (Exception e) {

        }


        t = new JTextArea(100,100);


     JScrollPane scroll=new JScrollPane(t);

     scroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SC
ROLLBAR_ALWAYS);

     scroll.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTA
L_SCROLLBAR_ALWAYS);

     JPanel statusBar = new JPanel(new BorderLayout());

     JLabel lines=new JLabel("Lines");

     final JLabel status = new JLabel("Statusbar");

        statusBar.add(status,BorderLayout.EAST);

     statusBar.add(lines,BorderLayout.WEST);


        JMenuBar mb = new JMenuBar();

        JMenu m1 = new JMenu("File");


        JMenuItem mi1 = new JMenuItem("New",new
ImageIcon("F:\\study\\java project\\img\\new.png"));
```

```
        JMenuItem mi2 = new JMenuItem("Open",new
ImageIcon("F:\\study\\java project\\img\\Open.png"));

        JMenuItem mi3 = new JMenuItem("Save",new
ImageIcon("F:\\study\\java project\\img\\Save.png"));

        JMenuItem mi9 = new JMenuItem("close",new
ImageIcon("F:\\study\\java project\\img\\close.png"));


        mi1.addActionListener(this);

        mi2.addActionListener(this);

        mi3.addActionListener(this);

        mi9.addActionListener(this);


        m1.add(mi1);

        m1.add(mi2);

        m1.add(mi3);

        m1.add(mi9);


        JMenu m2 = new JMenu("Edit");


        JMenuItem mi4 = new JMenuItem("cut",new
ImageIcon("F:\\study\\java project\\img\\Cut.png"));

        JMenuItem mi5 = new JMenuItem("copy",new
ImageIcon("F:\\study\\java project\\img\\Copy.png"));

        JMenuItem mi6 = new JMenuItem("paste",new
ImageIcon("F:\\study\\java project\\img\\Paste.png"));

     JMenuItem mi7 = new JMenuItem("print",new
ImageIcon("F:\\study\\java project\\img\\Print.png"));


        mi4.addActionListener(this);

        mi5.addActionListener(this);
```

```
         mi6.addActionListener(this);

      mi7.addActionListener(this);


         m2.add(mi4);

         m2.add(mi5);

         m2.add(mi6);

      m2.add(mi7);


      JMenu m3 = new JMenu("Format");


         JMenuItem mi8 = new JMenuItem("Font",new
ImageIcon("F:\\study\\java project\\img\\fonttt.png"));

      mi8.addActionListener(this);


         m3.add(mi8);


         JMenuItem mc = new JMenuItem("close");

      JMenu mr = new JMenu("Run");

      JMenuItem mir = new JMenuItem("Run java",new
ImageIcon("F:\\study\\java project\\img\\Run.png"));

      JMenuItem mirr = new JMenuItem("Run Applet",new
ImageIcon("F:\\study\\java project\\img\\Run.png"));

      mr.add(mir);

      mr.add(mirr);

         mc.addActionListener(this);

      mir.addActionListener(this);

      mirr.addActionListener(this);
```

```
        mb.add(m1);

        mb.add(m2);

     mb.add(m3);

     mb.add(mr);





DefaultMutableTreeNode node1=new DefaultMutableTreeNode
("Java");


DefaultMutableTreeNode inner1=new
DefaultMutableTreeNode("Loops");

DefaultMutableTreeNode inneri1=new
DefaultMutableTreeNode("while(conditon){\n\n }");

DefaultMutableTreeNode inneri2=new
DefaultMutableTreeNode("for( ){\n\n }");

DefaultMutableTreeNode inneri3=new
DefaultMutableTreeNode("do{\n\n }\nwhile(conditon)");


DefaultMutableTreeNode inner2=new
DefaultMutableTreeNode("Conditional and Return Statements");

DefaultMutableTreeNode inneri4=new
DefaultMutableTreeNode("if(Conditon){\n\n }");

DefaultMutableTreeNode inneri5=new
DefaultMutableTreeNode("else{\n\n }");

DefaultMutableTreeNode inneri6=new
DefaultMutableTreeNode("if(conditon){\n\n }\n else{\n\n }");

DefaultMutableTreeNode inneri7=new
DefaultMutableTreeNode("else if(conditon){\n\n }");
```

```
DefaultMutableTreeNode in1=new
DefaultMutableTreeNode("try{\n\n }\n catch(Exception e){\n\n
}");

DefaultMutableTreeNode in2=new
DefaultMutableTreeNode("try{\n\n }\n catch(Exception e){\n\n
}\n finally{\n\n }");




DefaultMutableTreeNode inner3=new
DefaultMutableTreeNode("Functions");


DefaultMutableTreeNode innerinner1=new
DefaultMutableTreeNode("String Functions");

DefaultMutableTreeNode in3=new
DefaultMutableTreeNode("compareTo()");

DefaultMutableTreeNode in4=new
DefaultMutableTreeNode("equals()");

DefaultMutableTreeNode in5=new
DefaultMutableTreeNode("valueOf()");

DefaultMutableTreeNode in6=new
DefaultMutableTreeNode("substring()");

DefaultMutableTreeNode in7=new
DefaultMutableTreeNode("replace()");

DefaultMutableTreeNode in8=new
DefaultMutableTreeNode("toString()");

DefaultMutableTreeNode in9=new
DefaultMutableTreeNode("charAt()");

DefaultMutableTreeNode in10=new
DefaultMutableTreeNode("concat()");

DefaultMutableTreeNode in11=new
DefaultMutableTreeNode("endsWith()");

DefaultMutableTreeNode in12=new
DefaultMutableTreeNode("lastIndexOf()");
```

```
DefaultMutableTreeNode in13=new
DefaultMutableTreeNode("firstIndexOf()");

DefaultMutableTreeNode in14=new
DefaultMutableTreeNode("getChars()");

DefaultMutableTreeNode in15=new
DefaultMutableTreeNode("getBytes()");

DefaultMutableTreeNode in16=new
DefaultMutableTreeNode("indexOf()");

DefaultMutableTreeNode in17=new
DefaultMutableTreeNode("length()");

DefaultMutableTreeNode in18=new
DefaultMutableTreeNode("split()");

DefaultMutableTreeNode in19=new
DefaultMutableTreeNode("startsWith()");

DefaultMutableTreeNode in20=new
DefaultMutableTreeNode("toLowerCase()");

DefaultMutableTreeNode in21=new
DefaultMutableTreeNode("toUpperCase()");


DefaultMutableTreeNode innerinner2=new
DefaultMutableTreeNode("Math Functions");

DefaultMutableTreeNode in22=new
DefaultMutableTreeNode("pow()");

DefaultMutableTreeNode in23=new
DefaultMutableTreeNode("sqrt()");

DefaultMutableTreeNode in24=new
DefaultMutableTreeNode("abs()");

DefaultMutableTreeNode in25=new
DefaultMutableTreeNode("ceil()");

DefaultMutableTreeNode in26=new
DefaultMutableTreeNode("floor()");

DefaultMutableTreeNode in27=new
DefaultMutableTreeNode("floorDiv()");
```

```
DefaultMutableTreeNode in28=new
DefaultMutableTreeNode("min()");

DefaultMutableTreeNode in29=new
DefaultMutableTreeNode("max()");

DefaultMutableTreeNode in30=new
DefaultMutableTreeNode("round()");

DefaultMutableTreeNode in31=new
DefaultMutableTreeNode("random()");




DefaultMutableTreeNode innerinner3=new
DefaultMutableTreeNode("Parsing Functions");

DefaultMutableTreeNode in32=new
DefaultMutableTreeNode("Integer.parseInt()");

DefaultMutableTreeNode in33=new
DefaultMutableTreeNode("Flot.parseFloat()");

DefaultMutableTreeNode in34=new
DefaultMutableTreeNode("Double.parseDouble()");

DefaultMutableTreeNode in35=new
DefaultMutableTreeNode("Byte.praseByte()");


DefaultMutableTreeNode innerinner4=new
DefaultMutableTreeNode("AWT Functions");

DefaultMutableTreeNode in36=new
DefaultMutableTreeNode("setSize()");

DefaultMutableTreeNode in37=new
DefaultMutableTreeNode("setVisible()");

DefaultMutableTreeNode in38=new
DefaultMutableTreeNode("setTitle()");

DefaultMutableTreeNode in39=new
DefaultMutableTreeNode("setText()()");

DefaultMutableTreeNode in40=new
DefaultMutableTreeNode("getText()");
```

```
DefaultMutableTreeNode in41=new
DefaultMutableTreeNode("setAlignment()");

DefaultMutableTreeNode in42=new
DefaultMutableTreeNode("getAlignment()");

DefaultMutableTreeNode in43=new
DefaultMutableTreeNode("setLabel()");

DefaultMutableTreeNode in44=new
DefaultMutableTreeNode("getLabel()");

DefaultMutableTreeNode in45=new
DefaultMutableTreeNode("setState()");

DefaultMutableTreeNode in46=new
DefaultMutableTreeNode("getState()");

DefaultMutableTreeNode in47=new
DefaultMutableTreeNode("getSelectedCheckBox()");

DefaultMutableTreeNode in48=new
DefaultMutableTreeNode("setSelectedCheckBox()");

DefaultMutableTreeNode in49=new
DefaultMutableTreeNode("add()");

DefaultMutableTreeNode in50=new
DefaultMutableTreeNode("getSelectedItem()");

DefaultMutableTreeNode in51=new
DefaultMutableTreeNode("getSelectedItems()");

DefaultMutableTreeNode in52=new
DefaultMutableTreeNode("select()");

DefaultMutableTreeNode in53=new
DefaultMutableTreeNode("getItemCount()");

DefaultMutableTreeNode in54=new
DefaultMutableTreeNode("getItem()");

DefaultMutableTreeNode in55=new
DefaultMutableTreeNode("getSelectedText()");

DefaultMutableTreeNode in56=new
DefaultMutableTreeNode("isEditable");

DefaultMutableTreeNode in57=new
DefaultMutableTreeNode("apped()");
```

```
DefaultMutableTreeNode in58=new
DefaultMutableTreeNode("insert()");

DefaultMutableTreeNode in59=new
DefaultMutableTreeNode("getEchoChar()");

DefaultMutableTreeNode in60=new
DefaultMutableTreeNode("setEchoChar()");

DefaultMutableTreeNode in61=new
DefaultMutableTreeNode("replaceRange()");




DefaultMutableTreeNode innerinner5=new
DefaultMutableTreeNode("Applet Life Cycle Methods");

DefaultMutableTreeNode app1=new DefaultMutableTreeNode("public
void init()\n{\n\n}");

DefaultMutableTreeNode app2=new DefaultMutableTreeNode("public
void start()\n{\n\n}");

DefaultMutableTreeNode app3=new DefaultMutableTreeNode("public
void paint()\n{\n\n}");

DefaultMutableTreeNode app4=new DefaultMutableTreeNode("public
void stop()\n{\n\n}");

DefaultMutableTreeNode app5=new DefaultMutableTreeNode("public
void destroy()\n{\n\n}");

DefaultMutableTreeNode app6=new
DefaultMutableTreeNode("repaint()");




DefaultMutableTreeNode innerinner6=new
DefaultMutableTreeNode("Other Functions");

DefaultMutableTreeNode inneri8=new
DefaultMutableTreeNode("System.out.println()");

DefaultMutableTreeNode inneri9=new
DefaultMutableTreeNode("public static void main(String
ar[]){\n\n }");
```

```
DefaultMutableTreeNode in62=new
DefaultMutableTreeNode("finalize()");




DefaultMutableTreeNode inner4=new
DefaultMutableTreeNode("Packages");

DefaultMutableTreeNode inneri10=new
DefaultMutableTreeNode("java.io");

DefaultMutableTreeNode inneri11=new
DefaultMutableTreeNode("java.util");

DefaultMutableTreeNode inneri12=new
DefaultMutableTreeNode("java.awt");

DefaultMutableTreeNode inneri13=new
DefaultMutableTreeNode("java.net");

DefaultMutableTreeNode inneri14=new
DefaultMutableTreeNode("java.awt.event");

DefaultMutableTreeNode in63=new
DefaultMutableTreeNode("java.lang");

DefaultMutableTreeNode in64=new
DefaultMutableTreeNode("java.applet");

DefaultMutableTreeNode inneri15=new
DefaultMutableTreeNode("javax.swing");

DefaultMutableTreeNode in65=new
DefaultMutableTreeNode("javax.swing.event");

DefaultMutableTreeNode in66=new
DefaultMutableTreeNode("javax.servlet");

DefaultMutableTreeNode in67=new
DefaultMutableTreeNode("javax.servlet.jsp");




DefaultMutableTreeNode inner6=new DefaultMutableTreeNode("AWT
and Swing Components");
```

```
DefaultMutableTreeNode inneri17=new
DefaultMutableTreeNode("Label");

DefaultMutableTreeNode inneri18=new
DefaultMutableTreeNode("Button");

DefaultMutableTreeNode inneri19=new
DefaultMutableTreeNode("Choice");

DefaultMutableTreeNode inneri21=new
DefaultMutableTreeNode("TextField");

DefaultMutableTreeNode inneri22=new
DefaultMutableTreeNode("TextArea");

DefaultMutableTreeNode inneri23=new
DefaultMutableTreeNode("Checkbox");

DefaultMutableTreeNode in68=new
DefaultMutableTreeNode("CheckBoxGroup");

DefaultMutableTreeNode in69=new
DefaultMutableTreeNode("List");

DefaultMutableTreeNode innerii25=new
DefaultMutableTreeNode("MenuBar");

DefaultMutableTreeNode innerii26=new
DefaultMutableTreeNode("Menu");

DefaultMutableTreeNode innerii27=new
DefaultMutableTreeNode("MenuItem");

DefaultMutableTreeNode in70=new
DefaultMutableTreeNode("JLabel");

DefaultMutableTreeNode in71=new
DefaultMutableTreeNode("JButton");

DefaultMutableTreeNode in72=new
DefaultMutableTreeNode("JTextField");

DefaultMutableTreeNode in73=new
DefaultMutableTreeNode("JTextArea");

DefaultMutableTreeNode in74=new
DefaultMutableTreeNode("JComboBox");

DefaultMutableTreeNode in75=new
DefaultMutableTreeNode("JCheckBox");
```

```
DefaultMutableTreeNode in76=new
DefaultMutableTreeNode("jRadioButton");

DefaultMutableTreeNode in77=new
DefaultMutableTreeNode("JTree");

DefaultMutableTreeNode in78=new
DefaultMutableTreeNode("JPasswordField");

DefaultMutableTreeNode in79=new
DefaultMutableTreeNode("JMenuBar");

DefaultMutableTreeNode in80=new
DefaultMutableTreeNode("JMenu");

DefaultMutableTreeNode in81=new
DefaultMutableTreeNode("JMenuItem");


node1.add(inner1);

inner1.add(inneri1);

inner1.add(inneri2);

inner1.add(inneri3);


node1.add(inner2);

inner2.add(inneri4);

inner2.add(inneri5);

inner2.add(inneri6);

inner2.add(inneri7);

inner2.add(in1);

inner2.add(in2);


node1.add(inner3);

inner3.add(innerinner1);

innerinner1.add(in3);

innerinner1.add(in4);
```

```
innerinner1.add(in5);

innerinner1.add(in6);

innerinner1.add(in7);

innerinner1.add(in8);

innerinner1.add(in9);

innerinner1.add(in10);

innerinner1.add(in11);

innerinner1.add(in12);

innerinner1.add(in13);

innerinner1.add(in14);

innerinner1.add(in15);

innerinner1.add(in16);

innerinner1.add(in17);

innerinner1.add(in18);

innerinner1.add(in19);

innerinner1.add(in20);

innerinner1.add(in21);



inner3.add(innerinner2);

innerinner2.add(in22);

innerinner2.add(in23);

innerinner2.add(in24);

innerinner2.add(in25);

innerinner2.add(in26);

innerinner2.add(in27);

innerinner2.add(in28);
```

```
innerinner2.add(in29);

innerinner2.add(in30);

innerinner2.add(in31);



inner3.add(innerinner3);

innerinner3.add(in32);

innerinner3.add(in33);

innerinner3.add(in34);

innerinner3.add(in35);


inner3.add(innerinner4);

innerinner4.add(in36);

innerinner4.add(in37);

innerinner4.add(in38);

innerinner4.add(in39);

innerinner4.add(in40);

innerinner4.add(in41);

innerinner4.add(in42);

innerinner4.add(in43);

innerinner4.add(in44);

innerinner4.add(in45);

innerinner4.add(in46);

innerinner4.add(in47);

innerinner4.add(in48);

innerinner4.add(in49);

innerinner4.add(in50);
```

```
innerinner4.add(in51);

innerinner4.add(in52);

innerinner4.add(in53);

innerinner4.add(in54);

innerinner4.add(in55);

innerinner4.add(in56);

innerinner4.add(in57);

innerinner4.add(in58);

innerinner4.add(in59);

innerinner4.add(in60);

innerinner4.add(in61);



inner3.add(innerinner5);

innerinner5.add(app1);

innerinner5.add(app2);

innerinner5.add(app3);

innerinner5.add(app4);

innerinner5.add(app5);

innerinner5.add(app6);



inner3.add(innerinner6);

innerinner6.add(inneri8);

innerinner6.add(inneri9);

innerinner6.add(in62);
```

```
node1.add(inner4);
inner4.add(inneri10);
inner4.add(inneri11);
inner4.add(inneri12);
inner4.add(inneri13);
inner4.add(inneri14);
inner4.add(in63);
inner4.add(in64);
inner4.add(inneri15);
inner4.add(in65);
inner4.add(in66);
inner4.add(in67);


node1.add(inner6);
inner6.add(inneri18);
inner6.add(inneri19);
inner6.add(inneri21);
inner6.add(inneri22);
inner6.add(inneri23);
inner6.add(in68);
inner6.add(in69);
inner6.add(innerii25);
inner6.add(innerii26);
inner6.add(innerii27);
inner6.add(in70);
```

```
inner6.add(in71);

inner6.add(in72);

inner6.add(in73);

inner6.add(in74);

inner6.add(in75);

inner6.add(in76);

inner6.add(in77);

inner6.add(in78);

inner6.add(in79);

inner6.add(in80);

inner6.add(in81);




JTree jt=new JTree(node1);

jt.setShowsRootHandles(true);

JScrollPane scroll1=new JScrollPane(jt);

scroll1.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLL
BAR_ALWAYS);

scroll1.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SC
ROLLBAR_ALWAYS);

f.add(scroll1,BorderLayout.WEST);

jt.getSelectionModel().addTreeSelectionListener(new
TreeSelectionListener(){

public void valueChanged(TreeSelectionEvent e){

DefaultMutableTreeNode selectedNode=(DefaultMutableTreeNode)
jt.getLastSelectedPathComponent();

try{

nodename=selectedNode.getUserObject().toString();
}

catch(Exception ee)

{



}

if(nodename=="Java"||nodename=="Loops"||nodename=="Conditional
and Return
Statements"||nodename=="Functions"||nodename=="Packages"||

nodename=="AWT and Swing Components"||nodename=="String
Functions"||nodename=="Math Functions"||nodename=="Parsing
Functions"||

nodename=="AWT Functions"||nodename=="Applet Life Cycle
Methods"||nodename=="Other Functions"){}

else{

t.insert(nodename,t.getCaretPosition());

}

}

});


        f.setJMenuBar(mb);

        f.add(scroll,BorderLayout.CENTER);

     f.add(statusBar, BorderLayout.SOUTH);

        f.setSize(1000, 600);

        f.show();

     f.addComponentListener(new ComponentAdapter() {


      public void componentResized(ComponentEvent e) {

        status.setText("SIZE:"+f.getWidth() + "x" +
f.getHeight());

      }
```

```
    });
     t.addKeyListener(new KeyAdapter(){
     public void keyReleased(KeyEvent e){
     int lineNumber=0, column=0, pos=0;
     try
     {
     pos=t.getCaretPosition();
     lineNumber=t.getLineOfOffset(pos);
     column=pos-t.getLineStartOffset(lineNumber);
     }catch(Exception excp){}
     if(t.getText().length()==0){lineNumber=0; column=0;}
     lines.setText("||   Line "+(lineNumber+1)+" , Column 
"+(column+1)+"   ||");
     }
     });



    }


    public void actionPerformed(ActionEvent e)
    {
        String s = e.getActionCommand();


        if (s.equals("cut")) {
            t.cut();
        }
        else if (s.equals("copy")) {
            t.copy();
        }
        else if (s.equals("paste")) {
            t.paste();
        }
        else if (s.equals("Save")) {
         int r=0;
         File fi=null;
            JFileChooser j = new JFileChooser("f:");
         if(name==null){
            r = j.showSaveDialog(null);
         }

            if (r == JFileChooser.APPROVE_OPTION) {

          if(name==null){
                fi = new
File(j.getSelectedFile().getAbsolutePath());
          name=fi.getName();
          f.setTitle("Editor-"+name);
          path=j.getSelectedFile().getAbsolutePath();
          }
          else{
          fi=new File(path);
          }

                try {
                    FileWriter wr = new FileWriter(fi, false);
                    BufferedWriter w = new BufferedWriter(wr);


                    w.write(t.getText());


                    w.flush();

                    w.close();

              JOptionPane.showMessageDialog(f, "File Saved");

                }

                catch (Exception evt) {

                    JOptionPane.showMessageDialog(f,
evt.getMessage());

                }

            }


            else

                JOptionPane.showMessageDialog(f, "the user
cancelled the operation");

        }

        else if (s.equals("print")) {

            try {

                t.print();

            }

            catch (Exception evt) {

                JOptionPane.showMessageDialog(f,
evt.getMessage());

            }

        }

        else if (s.equals("Open")) {
            JFileChooser j = new JFileChooser("f:");

            int r = j.showOpenDialog(null);

            if (r == JFileChooser.APPROVE_OPTION) {

                File fi = new
File(j.getSelectedFile().getAbsolutePath());
          name=fi.getName();
          f.setTitle("Editor-"+name);
          path=j.getSelectedFile().getAbsolutePath();
                try {
                    String s1 = "", sl = "";

                    FileReader fr = new FileReader(fi);

                    BufferedReader br = new
BufferedReader(fr);

                    sl = br.readLine();

                    while ((s1 = br.readLine()) != null) {
                        sl = sl + "\n" + s1;
                    }

                    t.setText(sl);
                }
                catch (Exception evt) {
                    JOptionPane.showMessageDialog(f,
evt.getMessage());
                }
            }
            else
                JOptionPane.showMessageDialog(f, "the user
cancelled the operation");
        }
        else if (s.equals("New")) {
            t.setText("");
         name=null;
         f.setTitle("Editor-Untitled");
        }
        else if (s.equals("close")) {
            f.setVisible(false);
        }

     else if(s.equals("Font")){
     m_font= FontChooser.showDialog(this, "Select font");
     t.setFont(m_font);
     }

     else if (s.equals("Run java")) {
         File ff=new File("C:\\Users\\admin\\fille.bat");
     try{
         String s1 = "", sl = "",temp1,temp2;
         int flag=1;
```

```
           FileReader fr = new FileReader(ff);
        BufferedReader br = new BufferedReader(fr);
           s1 = br.readLine();
        temp1=s1.substring(0,1);
        temp2=path.substring(0,1);
        s1=s1.replace(temp1,temp2);



           while ((s1 = br.readLine()) != null) {
         if(flag==1){
         temp1=s1.substring(3,s1.lastIndexOf("\\"));
         temp2=path.substring(0,path.lastIndexOf("\\"));
         s1=s1.replace(temp1,temp2);
         }
         if(flag==2){
         temp1=s1.substring(6,s1.lastIndexOf("."));
         s1=s1.replace(temp1+".java",name);
         }
         if(flag==3){

    s1=s1.replace(temp1,name.substring(0,name.lastIndexOf(".")
)));

         }
               s1 = s1 + "\n" + s1;
         flag++;
           }
        br.close();
           FileWriter wr = new FileWriter(ff, false);
```

```
            BufferedWriter w = new BufferedWriter(wr);

            w.write(sl);

            w.flush();

            w.close();

         Desktop.getDesktop().open(ff);

        }

     catch(Exception evt){

      JOptionPane.showMessageDialog(f, "Please Select a File
First");

     }


     }



     else if (s.equals("Run Applet")) {

         File ff=new File("C:\\Users\\admin\\fille2.bat");

     try{

         String s1 = "", sl = "",temp1,temp2;

         int flag=1;

            FileReader fr = new FileReader(ff);

         BufferedReader br = new BufferedReader(fr);

            sl = br.readLine();

         temp1=sl.substring(0,1);

         temp2=path.substring(0,1);

         sl=sl.replace(temp1,temp2);



            while ((s1 = br.readLine()) != null) {
```

```
          if(flag==1){
          temp1=s1.substring(3,s1.lastIndexOf("\\"));
          temp2=path.substring(0,path.lastIndexOf("\\"));
          s1=s1.replace(temp1,temp2);
          }
          if(flag==2){
          temp1=s1.substring(6,s1.lastIndexOf("."));
          s1=s1.replace(temp1+".java",name);
          }
          if(flag==3){
          s1=s1.replace(temp1+".java",name);
          }
                sl = sl + "\n" + s1;
          flag++;
            }
         br.close();
            FileWriter wr = new FileWriter(ff, false);
            BufferedWriter w = new BufferedWriter(wr);
            w.write(sl);
            w.flush();
            w.close();
         Desktop.getDesktop().open(ff);
        }
     catch(Exception evt){
      JOptionPane.showMessageDialog(f, "Please Select a File
First");
     }
```

```
      }

   }

    public static void main(String args[])
    {
        Editor e = new Editor();
    }
}

class FontChooser extends JComponent
{

  public static Font showDialog(Component parent, String
title)
  {

    final FontChooser pane = new FontChooser();

    FontTracker ok = new FontTracker(pane);

    JDialog dialog =
        createDialog(parent, title, true,
                     pane, ok, null
        );
     dialog.setVisible(true);
    dialog.addWindowListener(
        new FontChooserDialog.Closer());
    dialog.addComponentListener(
        new FontChooserDialog.DisposeOnClose());

    return ok.getSelectedFont();

  }

  public static JDialog createDialog(
            Component parent, String title,
            boolean modal,
            FontChooser chooserPane,
            ActionListener okListener,
            ActionListener cancelListener)
  {
    return new FontChooserDialog(
            parent, title, modal, chooserPane,
            okListener, cancelListener);
  }

  public FontChooser()
  {
    setLayout(new BorderLayout());

    PreviewPanel previewPane = new PreviewPanel();
    m_inputPane = new InputPanel(previewPane);
    add(m_inputPane, BorderLayout.CENTER);
    add(previewPane, BorderLayout.SOUTH);
  }

  public Font getSelectedFont()
  {
    return m_inputPane.getSelectedFont();
  }

  private InputPanel m_inputPane;

  class InputPanel extends JPanel
  {
    public InputPanel(ListSelectionListener listener)
    {
      setLayout(new BorderLayout());

      Box nameBox = Box.createVerticalBox();
      nameBox.add(Box.createVerticalStrut(10));
      JLabel fontNameLabel = new JLabel("Font Name:");
      nameBox.add(fontNameLabel);
      if (listener != null)
      {
        m_fontNameList.addListSelectionListener(listener);
      }
      JScrollPane namePane =
                    new JScrollPane(m_fontNameList);
      nameBox.add(namePane);
      nameBox.add(Box.createVerticalStrut(10));

      Box styleBox = Box.createVerticalBox();
      styleBox.add(Box.createVerticalStrut(10));
      JLabel fontStyleLabel = new JLabel("Font Style:");
      styleBox.add(fontStyleLabel);
      if (listener != null)
      {
        m_fontStyleList.addListSelectionListener(listener);
      }
      JScrollPane stylePane = new
JScrollPane(m_fontStyleList);
      styleBox.add(stylePane);
      styleBox.add(Box.createVerticalStrut(10));

      Box sizeBox = Box.createVerticalBox();
      sizeBox.add(Box.createVerticalStrut(10));
      JLabel fontSizeLabel = new JLabel("Size:");
      sizeBox.add(fontSizeLabel);
      if (listener != null)
      {
        m_fontSizeList.addListSelectionListener(listener);
      }
      JScrollPane sizePane = new JScrollPane(m_fontSizeList);
      sizeBox.add(sizePane);
      sizeBox.add(Box.createVerticalStrut(10));
```

```
    Box mainBox = Box.createHorizontalBox();
    mainBox.add(Box.createHorizontalStrut(10));
    mainBox.add(nameBox);
    mainBox.add(Box.createHorizontalStrut(10));
    mainBox.add(styleBox);
    mainBox.add(Box.createHorizontalStrut(10));
    mainBox.add(sizeBox);
    mainBox.add(Box.createHorizontalStrut(10));
    add(mainBox, BorderLayout.CENTER);
  }

  public Font getSelectedFont()
  {
    return new Font(m_fontNameList.getFontName(),
                    m_fontStyleList.getFontStyle(),
                    m_fontSizeList.getFontSize()
        );
  }

  private FontNameList m_fontNameList =
                          new FontNameList();
  private FontStyleList m_fontStyleList =
                          new FontStyleList();
  private FontSizeList m_fontSizeList =
                          new FontSizeList();
}
```

```
class PreviewPanel extends JPanel
    implements ListSelectionListener
{

  public PreviewPanel()
  {
    setLayout(new FlowLayout());

    Box box = Box.createVerticalBox();
    JLabel previewLabel = new JLabel("Preview:");
    box.add(previewLabel);
    m_text.setEditable(false);
    m_text.setBackground(Color.white);
    m_text.setForeground(Color.black);
    JScrollPane pane = new JScrollPane(m_text);
    pane.setPreferredSize(new Dimension(300, 80));
    box.add(pane);

    add(box);
  }

  public void valueChanged(ListSelectionEvent ev)
  {
    m_text.setFont(FontChooser.this.getSelectedFont());
  }

  private JTextField m_text = new JTextField(
        "This is sample text");
  }
}

class FontNameList extends JList
{

  FontNameList()
  {
    super(m_fontNames);
    setSelectedIndex(0);
    setVisibleRowCount(5);
  }

  String getFontName()
  {
    String name = (String) getSelectedValue();
    return name;
  }

  private static final String[] m_fontNames =
      GraphicsEnvironment.getLocalGraphicsEnvironment().
      getAvailableFontFamilyNames();
}

class FontStyleList extends JList
{
```

```
FontStyleList()

{

  super(m_fontStyles);

  setSelectedIndex(0);

  setVisibleRowCount(5);

}


int getFontStyle()

{

  int style = 0;

  String name = (String) getSelectedValue();

  if (name.equals("Regular"))

  {

    style = Font.PLAIN;

  }

  else if (name.equals("Italic"))

  {

    style = Font.ITALIC;

  }

  else if (name.equals("Bold"))

  {

    style = Font.BOLD;

  }

  else

  {

    style = Font.BOLD + Font.ITALIC;

  }
```

```
    return style;
  }

  private static final String[] m_fontStyles =
  {
    "Regular", "Italic", "Bold", "Bold Italic"
  };
}

class FontSizeList extends JList
{

  FontSizeList()
  {
    super(m_fontSizes);
    setSelectedIndex(4);
    setVisibleRowCount(5);
  }

  int getFontSize()
  {
    int size = Integer.parseInt(
                    (String) getSelectedValue());
    return size;
  }

  private static final String[] m_fontSizes =
  {
    "6", "8", "10", "12", "14", "16", "18",
    "20", "22", "24", "36", "72"
  };
}

class FontChooserDialog extends JDialog
{

  public FontChooserDialog(
                 Component component,
                 String title,
                 boolean modal,
                 FontChooser chooserPane,
                 ActionListener okListener,
                 ActionListener cancelListener)
  {
    super(JOptionPane.getFrameForComponent(component),
          title, modal);
      System.out.println("hello");

    m_chooserPane = chooserPane;

    Container contentPane = getContentPane();
    contentPane.setLayout(new BorderLayout());
    contentPane.add(m_chooserPane, BorderLayout.CENTER);
```

```
JPanel buttonPane = new JPanel();
buttonPane.setLayout(new FlowLayout(FlowLayout.CENTER));

JButton okButton = new JButton("OK");
buttonPane.add(okButton);
getRootPane().setDefaultButton(okButton);
okButton.setActionCommand("OK");
if (okListener != null)
{
  okButton.addActionListener(okListener);
}
okButton.addActionListener(new ActionListener()
{
  public void actionPerformed(ActionEvent e)
  {
    dispose();
  }
}
);

JButton cancelButton = new JButton("Cancel");
buttonPane.add(cancelButton);
cancelButton.setActionCommand("cancel");
if (cancelListener != null)
{
  cancelButton.addActionListener(cancelListener);
}
```

```
  cancelButton.addActionListener(new ActionListener()
  {
    public void actionPerformed(ActionEvent e)
    {
      dispose();
    }
  }
  );

  contentPane.add(buttonPane, BorderLayout.SOUTH);

  pack();
  setLocationRelativeTo(component);

}
static class Closer extends WindowAdapter
{
  public void windowClosing(WindowEvent e)
  {
    Window w = e.getWindow();
    w.setVisible(false);
  }
}

static class DisposeOnClose extends ComponentAdapter
{
  public void componentHidden(ComponentEvent e)
```

```
    {
      Window w = (Window) e.getComponent();
      w.dispose();
    }
  }

  private FontChooser m_chooserPane;
}

class FontTracker implements ActionListener
{
  public FontTracker(FontChooser chooser)
  {
    m_chooser = chooser;
  }

  public void actionPerformed(ActionEvent e)
  {
    m_font = m_chooser.getSelectedFont();
  }
  public Font getSelectedFont()
  {
    return m_font;
  }
  private FontChooser m_chooser;
  private Font m_font;
}
```

2) **fille.bat:-**

```
f:

cd f:\

javac MidLines.java

java MidLines

pause
```

3) **Fille2.bat:-**

```
f:

cd f:\

javac MidLines.java

appletviewer MidLines.java

pause
```

## Conclusion:-

In this way we have created a mini project i.e. “Java Editor” and written a report on it containing each and every detail of our project. Through this project we have learned different type concepts to create our project, we have also learned that how other editor works and what are the functions they provide in their editor. In this report we have covered the chapters such as

1) Introduction.
2) Features.
3) Advantages and Disadvantages.
4) Implementation.

We have explained that how our project works and its detailed information in the introduction. Other than that we have also defined that, what is the need for our project and scope for our project in the abstract.