

ABSTRACT

- We have developed a TEXT EDITOR Using Netbeans in Java.It exhibits the basic functionalities which any text editor should have but we have personalized.
- Netbeans was chosen because any editor requires a good Graphical User Interface and Netbeans provide that.
- The functions incorporate needed accurate syntaxs for proper functioning and this was done by java.
- Our next challenge was to know all syntaxes and put them in proper places in the source code.

INDEX

Sr No.	contents	Page No.
1.	introduction	3
2.	Software requirements	
3.	Overall Description	
4.	code	
5.	result	
6.	reference	

Introduction

Advance java programming:-

The history of Java is very interesting. Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time. The history of Java starts with the Green Team. Java team members (also known as Green Team), initiated this project to develop a language for digital devices such as set-top boxes, televisions, etc. However, it was best suited for internet programming. Later, Java technology was incorporated by Netscape.

The principles for creating Java programming were "Simple, Robust, Portable, Platform-independent, Secured, High Performance, Multithreaded, Architecture Neutral, Object-Oriented, Interpreted, and Dynamic". [Java](#) was developed by James Gosling, who is known as the father of Java, in 1995. James Gosling and his team members started the project in the early '90s.



- 1) Currently, Java is used in internet programming, mobile devices, games, e-business solutions, etc. Following are given significant points that describe the history of Java.
- 2) James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. The small team of sun engineers called Green Team.

Text editor:-

Text Editor is software that edits plain text and comprises of Java Swings and AWT. This project has all the frames prepared in Swing. These days, most of the programmers use Swing. A text editor is a type of computer program that edits plain text. Such programs are sometimes known as "notepad" software, following the naming of Microsoft Notepad. Text editors are provided with operating systems and software development packages, and can be used to change files such as configuration files, documentation files and programming language source code.

Editor: an editor is general-purpose system software, which can be used to enter and/or modify text, programs, graphics or data. An editor may be a line editor, screen editor, graphics editor or word processors.

Text Editor: A text editor is a type of program used for editing plain text files. Text editors are often used to change configuration files, documentation files, programming language source code, etc.



Types of editors:

Editor are broadly classified as:-

- 1.line Editors
- 2.Screen Editors
- 3.Graphic Editors

GUI: GUI stand for graphical user interface.it presents an easy to-use visual display to the user. It is made up of buttons,Labels,Windows,etc.

Software requirements:

Operating system	:	windows xp/7/8
Compiler used	:	java
IDE USED	:	Neatbeans
Front End	:	java, etc.

Overall Description

Product perspective: The product is a full screen text editor for the windows operating system with the capabilities to edit existing files and also create new ones.

File operations:-

1. Create a new file
- 2 .Open an existing file
3. Save and save as a file
4. Delete a file
5. Print a file

Editing operations:-

- 1.Cut copy and paste
2. Find and Replace

Code

```
import java.awt.*;
import javax.swing.*;
import java.io.*;
import java.awt.event.*;
import javax.swing.plaf.metal.*;
import javax.swing.text.*;

class editor extends JFrame implements ActionListener {

    // Text component
    JTextArea t;

    // Frame
    JFrame f;

    // Constructor
    editor()
    {
        // Create a frame
        f = new JFrame("editor");

        try {
            // Set metal look and feel
```

```
UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel")  
;
```

```
        // Set theme to ocean  
        MetalLookAndFeel.setCurrentTheme(new OceanTheme());  
    }  
    catch (Exception e) {  
    }
```

```
    // Text component  
    t = new JTextArea();
```

```
    // Create a menubar  
    JMenuBar mb = new JMenuBar();
```

```
    // Create a menu for menu  
    JMenu m1 = new JMenu("File");
```

```
    // Create menu items  
    JMenuItem mi1 = new JMenuItem("New");  
    JMenuItem mi2 = new JMenuItem("Open");  
    JMenuItem mi3 = new JMenuItem("Save");  
    JMenuItem mi9 = new JMenuItem("Print");
```

```
    // Add action listener  
    mi1.addActionListener(this);
```



```
mi2.addActionListener(this);  
mi3.addActionListener(this);
```

```
m1.add(mi1);  
m1.add(mi2);  
m1.add(mi3);  
m1.add(mi9);
```

```
// Create a menu for menu  
JMenu m2 = new JMenu("Edit");
```

```
// Create menu items  
JMenuItem mi4 = new JMenuItem("cut");  
JMenuItem mi5 = new JMenuItem("copy");  
JMenuItem mi6 = new JMenuItem("paste");
```

```
// Add action listener  
mi4.addActionListener(this);  
mi5.addActionListener(this);  
mi6.addActionListener(this);
```

```
m2.add(mi4);  
m2.add(mi5);  
m2.add(mi6);
```

```
JMenuItem mc = new JMenuItem("close");
```

```
mc.addActionListener(this);

mb.add(m1);
mb.add(m2);
mb.add(mc);

f.setJMenuBar(mb);
f.add(t);
f.setSize(500, 500);
f.show();
}

// If a button is pressed
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")) {  
    // Create an object of JFileChooser class  
    JFileChooser j = new JFileChooser("f:");  
  
    // Invoke the showsSaveDialog function to show the save  
dialog  
    int r = j.showSaveDialog(null);  
  
    if (r == JFileChooser.APPROVE_OPTION) {  
  
        // Set the label to the path of the selected directory  
        File fi = new  
File(j.getSelectedFile().getAbsolutePath());  
  
        try {  
            // Create a file writer  
            FileWriter wr = new FileWriter(fi, false);  
  
            // Create buffered writer to write  
            BufferedWriter w = new BufferedWriter(wr);  
  
            // Write  
            w.write(t.getText());  
  
            w.flush();  
            w.close();  
        }  
    }  
}
```

```

        }
        catch (Exception evt) {
            JOptionPane.showMessageDialog(f,
evt.getMessage());
        }
    }
    // If the user cancelled the operation
    else
        JOptionPane.showMessageDialog(f, "the user
cancelled the operation");
    }
    else if (s.equals("Print")) {
        try {
            // print the file
            t.print();
        }
        catch (Exception evt) {
            JOptionPane.showMessageDialog(f,
evt.getMessage());
        }
    }
    else if (s.equals("Open")) {
        // Create an object of JFileChooser class
        JFileChooser j = new JFileChooser("f:");

        // Invoke the showsOpenDialog function to show the save
dialog

```

```
int r = j.showOpenDialog(null);

// If the user selects a file
if (r == JFileChooser.APPROVE_OPTION) {

    // Set the label to the path of the selected directory
    File fi = new
File(j.getSelectedFile().getAbsolutePath());

    try {

        // String
        String s1 = "", sl = "";

        // File reader
        FileReader fr = new FileReader(fi);

        // Buffered reader
        BufferedReader br = new BufferedReader(fr);

        // Initialize sl
        sl = br.readLine();

        // Take the input from the file
        while ((s1 = br.readLine()) != null) {
            sl = sl + "\n" + s1;
        }
    }
}
```

```

        // Set the text
        t.setText(sl);
    }
    catch (Exception evt) {
        JOptionPane.showMessageDialog(f,
evt.getMessage());
    }
}
// If the user cancelled the operation
else
    JOptionPane.showMessageDialog(f, "the user
cancelled the operation");
}
else if (s.equals("New")) {
    t.setText("");
}
else if (s.equals("close")) {
    f.setVisible(false);
}
}

// Main class
public static void main(String args[])
{
    editor e = new editor();
}
}

```

Result



Fig.1. File

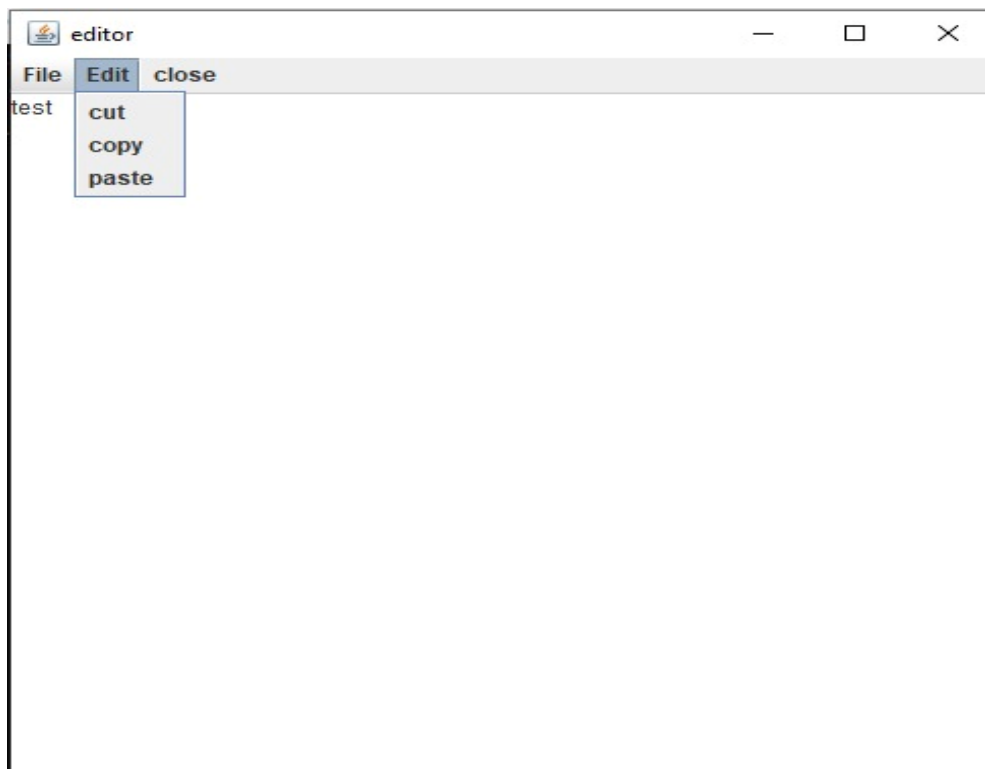


Fig.2. edit page

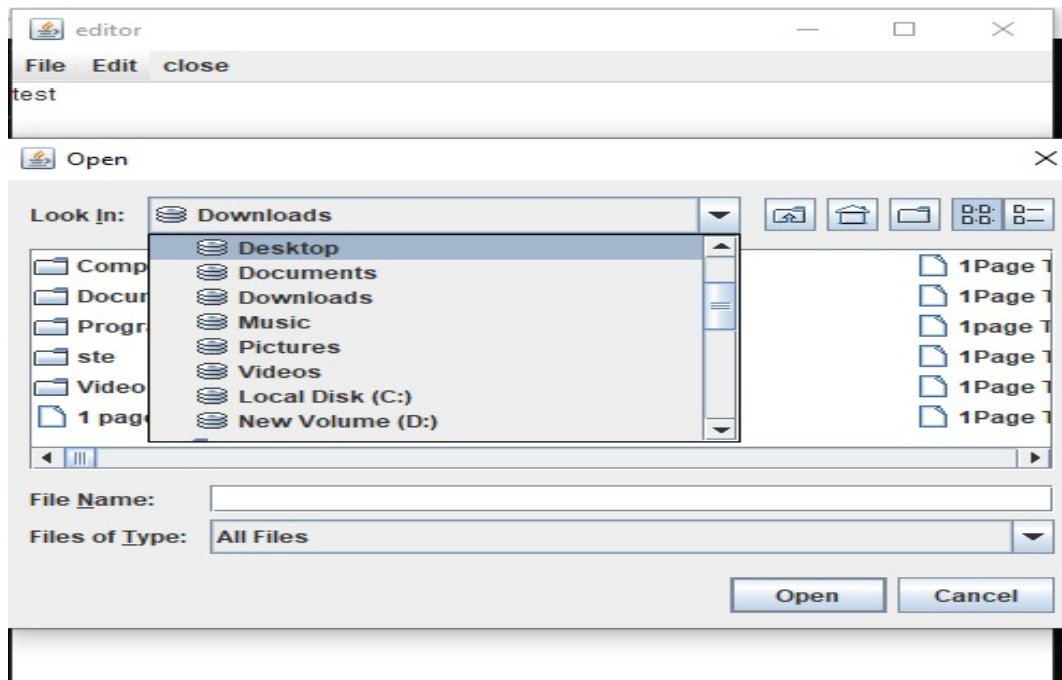


Fig.3. open page

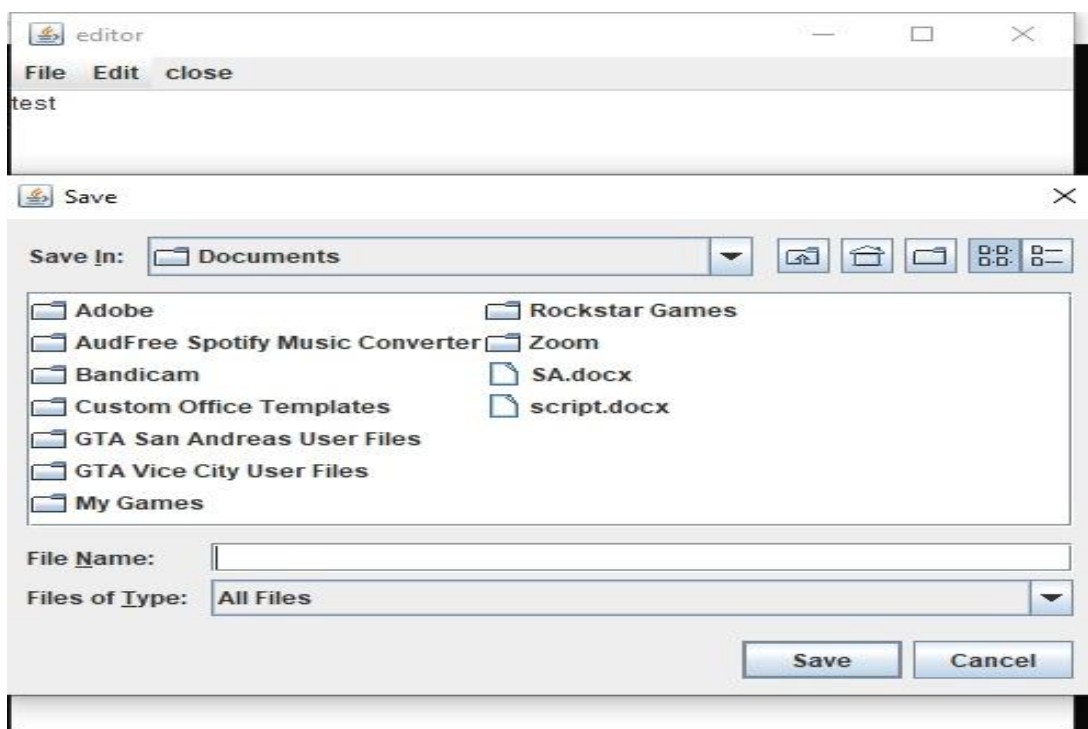


Fig.4. save page



Fig.5. cancel option

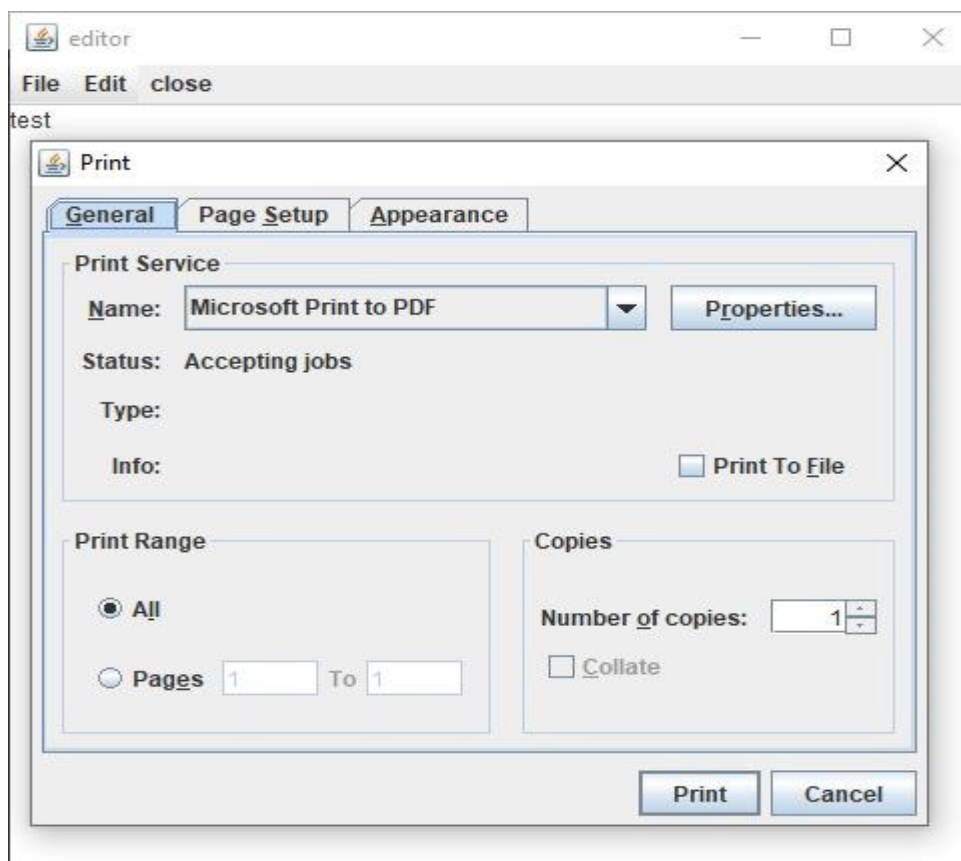


Fig.6. general page

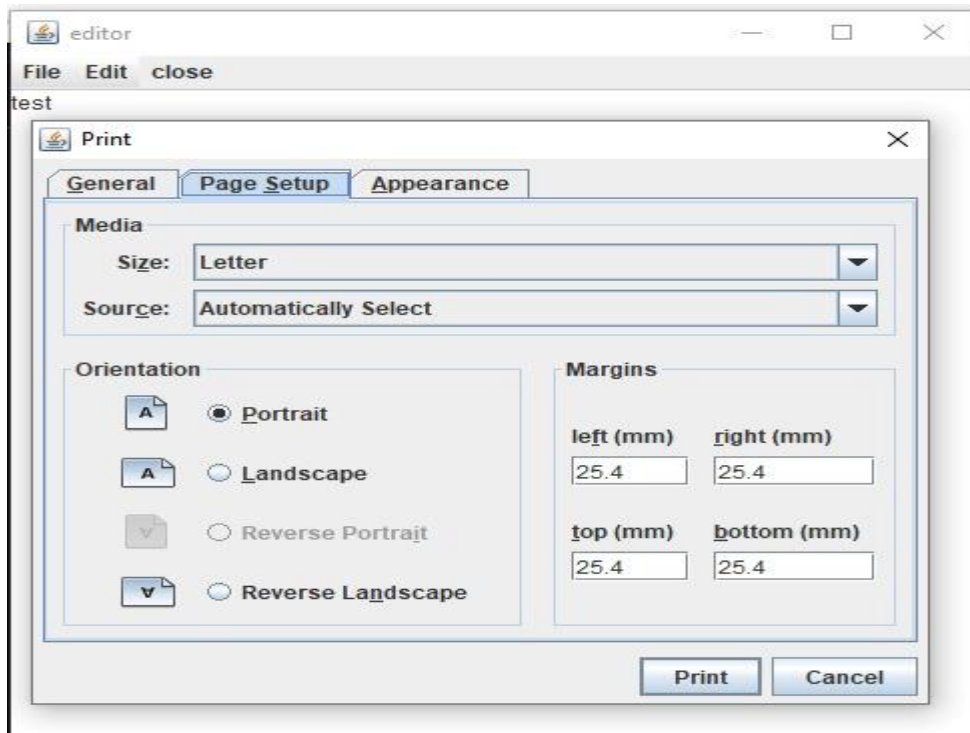


Fig.7. page setup

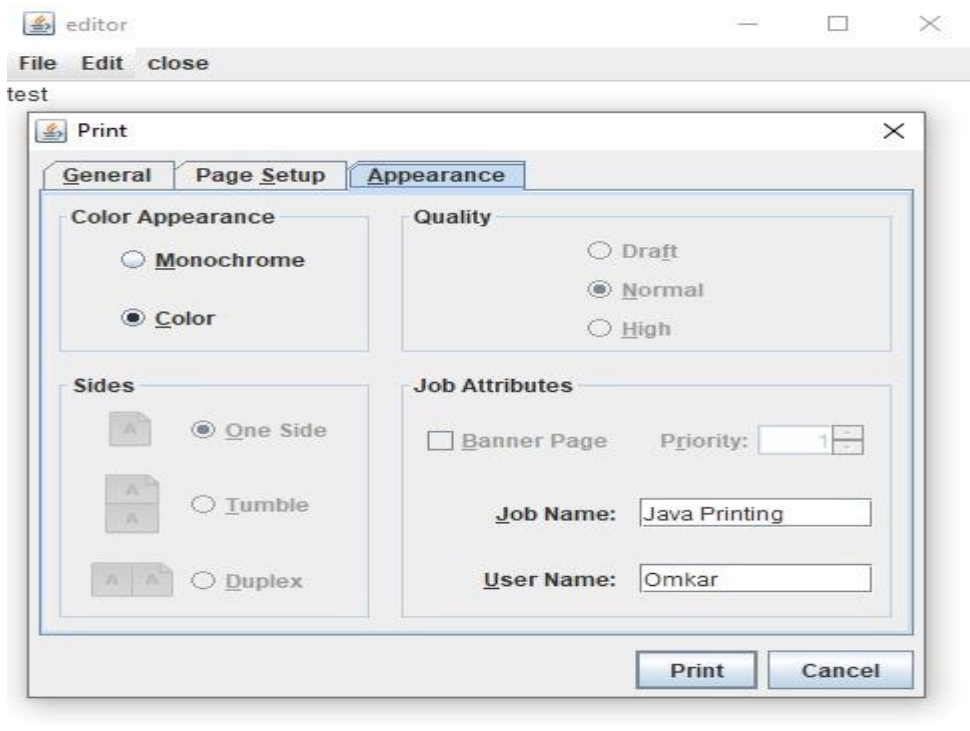


Fig.8. Appearance

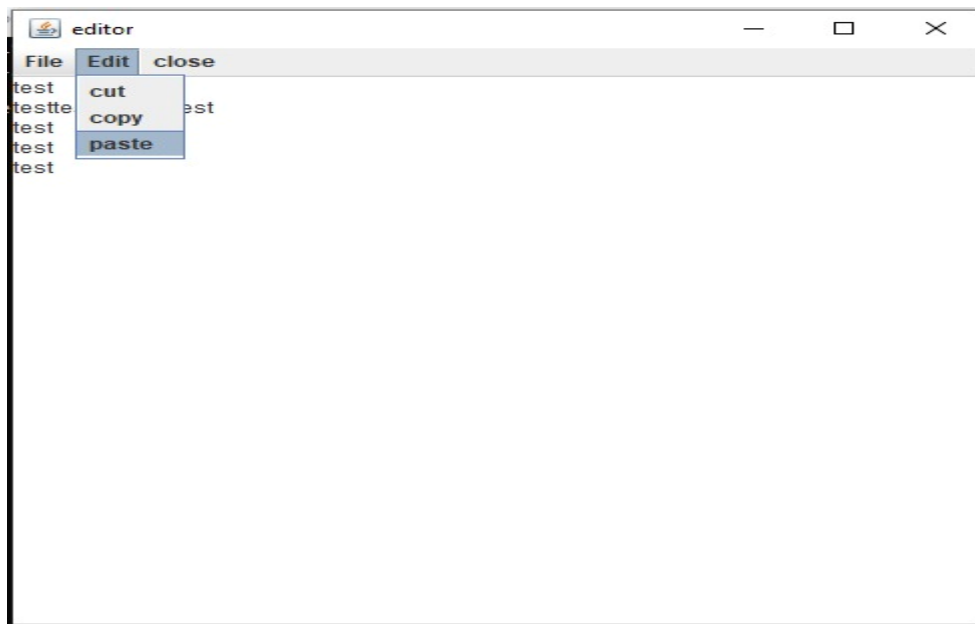


Fig.9. paste

Reference :-

1. <https://www.codecademy.com/catalog/language/java>
2. [Java Examples \(w3schools.com\)](https://www.w3schools.com/java/)
3. <https://www.youtube.com/>