**“SMARTPAD”**

**A MINOR PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD**

**OF**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**SUBMITTED TO**

## RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA

**BHOPAL (M.P)**

****

**Submitted by:**

**RUCHI SAXENA SHIVANGI GUPTA**

**0104CS111075 0104CS111087**

**KHUSHBOO BANSAL PUJA KUMARI**

**0104CS111036 0104CS111064**

**SWAMINI BHATT**

**0104CS111101**

**Guided By HOD**

**Mr. Niresh Sharma Mr. Gaurav Shrivastava**

**Mr. Abhijit Dwivedi (Asst. Professor)**

**(Asst. Professor)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

***RKDF INSTITUTE OF SCIENCE AND TECHNOLOGY***

***BHOPAL (M.P.)*APRIL 2014**

***RKDF INSTITUTE OF SCIENCE AND TECHNOLOGY***

***BHOPAL (M.P.)***

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CANDIDATES DECLARATION**

We hereby declare that the work, which is being presented in the major project entitled “***SMARTPAD***” in partial fulfillment of the requirements for the award of degree of Bachelor of Engineering in Computer Science and Engineering is an authentic record of work done by us.

**RUCHI SAXENA SHIVANGI GUPTA**

**0104CS111075 0104CS111087**

**KHUSHBO BANSAL PUJA KUMARI**

**0104CS111036 0104CS111064 SWAMINI BHATT**

**0104CS111101**

***RKDF INSTITUTE OF SCIENCE AND TECHNOLOGY***

***BHOPAL (M.P.)***

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that the project report on **“SMARTPAD”**submitted by RUCHI SAXENA(0104CS111075), SHIVANGI GUPTA(0104CS111087), KHUSHBOO BANSAL(0104CS111036), PUJA KUMARI(0104CS111064) & SWAIMINI BHATT(0104111101) to Department of Computer Science & Engineering, RKDFIST, Bhopal, in the partial fulfillment of the requirement of the award of the degree of **“Bachelor of Engineering(Computer Science & Engineering)”** is a satisfactory account of their project work and is recommended for their award of degree.

**Guided By Head OF DEPARTEMENT**

**Mr. Niresh Sharma**

**Mr. Gaurav Shrivastava**

**Mr. Abhijit Dwivedi**

**(Asst. Professor)**

**ACKNOWLEDGEMENT**

It is a privilege to express our deep sense of gratitude to our guide **Mr. Niresh Sharma &**

**Mr. Abhijit Dwivedi (Asst. Professor)** Department of Computer Science and Engineering, RKDFIST, Bhopal, M.P for his constant encouragement, valuable guidance and benevolent help, which was of greatest support to bring this work in its present shape. This work is the result of inspiration, support, guidance, motivation, cooperation and facilities that were extended to us at their best at all levels. The discussions with them regarding various issues of our project have been very beneficial and gave us a new direction of thinking. All these discussions have indeed played a vital role in progress of our work at many critical points during my endeavor.

We are highly indebted to **Mr. Gaurav Shrivastava** Head, Department of Computer Science & Engineering, RKDFIST, Bhopal for providing us all the necessary facilities and guidance.

We are thankful to our faculties for their valuable lectures in UML design, etc.which helped us in designing this project. We would also like to acknowledge ones, who, from behind the scenes have contributed their ideas and energies.

RUCHI SAXENA (0104CS111075)

SHIVANGI GUPTA (0104cs111087)

KHUSBOO BANSAL (0104CS111036)

PUJA KUMARI (0104CS111064)

SWAMINI BHATT (0104CS111101)

**TABLE OF CONTENTS FOR MINOR PROJECT**

**PAGE NO.**

**1. INTRODUCTION**------------------------------------------------------------------06

1.1 ABSTRACT -------------------------------------------------------------- 07

1.2 AIM------------------------------------------------------------------------- 08

1.3 ADVANTAGE / DISADVANTAGE---------------------------------- 08

**2. SYSTEM ANALYSIS** ---------------------------------------------------------- 092.1 REQUIREMENT ANALYSIS----------------------------------------- 09

2.2 NONFUNCTIONAL REQUIREMENTS----------------------------- 09

2.3 REQUIREMENT DEFINITION

AND SPECIFICATION------------------------------------------------- 10

2.4 HARDWARE REQUIREMENT-------------------------------------- 10

2.5 SOFTWARE REQUIREMENT--------------------------------------- 11

**3. TOOLS AND TECHNOLOGY**------------------------------------------------- 11

**4. PROCESS MODEL ADOPTED-----------------------------------------------** 13

**5. DESIGN------------------------------------------------------------------------------** 15 5.1 E-R DIAGRAM---------------------------------------------------------- 15

5.2 CLASS DIAGRAM-------------------------------------------------------16

**6. CODING------------------------------------------------------------------------------**19

**7. TESTING & DEBUGING---------------------------------------------------------**47

**8. SCREEN SHOTS--------------------------------------------------------------------**48

8.1 FEATURES-----------------------------------------------------------------48

8.2 OLD TECHNOLOGY v/s NEW TECHNOLOGY--------------------54

**9. FUTURE SCOPE OF THE PROPOSED SYSTEM -----------------------**55

**10. REFERENCES ---------------------------------------------------------------------**56

10.1 BIBLIOGRAPHY

10.2 WEBSITES REFERRENCED

**1. INTRODUCTION**

Notepad is a common text-only (plain text) editor. The resulting files—typically saved with the .txt extension—have no format tags or styles, making the program suitable for editing system file that are to be used in a dos environment.

Notepad supports both left-to-right and right-to-left based languages, and one can alternate between these viewing formats by using the right or left Ctrl+Shift keys to go to right-to-left format or left-to-right format, respectively.

Unlike,wordpad Notepad does not treat newlines in Unix- or Mac-style text files correctly.  
Early versions of Notepad offered only the most basic functions, such as finding text. Newer versions of Windows include an updated version of Notepad with a search and replace function (Ctrl + H), as well as Ctrl + F for search and similar keyboard shortcut.

Notepad makes use of a built-in window class named "EDIT". In older versions such as those included with Windows 95, Windows 98, there is a 64k limit on the size of the file being edited, an operating system limit of the EDIT class.

Up to Windows 95, Fixedsys was the only available font for Notepad. Windows NT 4.0 and 98 introduced the ability to change this font. As of Windows 2000, the default font was changed to Lucida Console. The font setting, however, only affects how the text is shown to the user and how it is printed, not how the file is saved to disk.

Up to Windows Me, there were almost no keyboard shortcuts and no line-counting feature. Starting with Windows 2000, shortcuts for common tasks like new, open and save were added, as well as a status-bar with a line counter.

Notepad also has a simple built-in logging function. Each time a file that starts with .LOG is opened, the program inserts a text timestamp on the last line of the file.  
Notepad accepts text from the Windows clipboard. When clipboard data with multiple formats is pasted into Notepad, the program will only accept text in the CF\_TEXT format. This is useful for stripping embedded font type and style codes from formatted text, such as when copying text from a web page and pasting into an email message or other WYSIWYG text editor. The formatted text is temporarily pasted into Notepad, and then immediately copied again in stripped format to bepasted into the Simple text editors like Notepad may be used to edit text with markup, such as HTML. However, because they lack many features (such as syntax highlighting), web developers may favor more specialized editors for this purpose.

1.1 **ABSTRACT**

As far as the problem domain is concerned, this project is basically used for the purpose of performing various editing task such as cut, copy, paste etc and many more advans features are also being included.

The princile working of this project is that in this project is that all the various features of a Editor are provided to the user in only a single software entity this allows the user to

decrease the cost of expenditure on this kind of software. This editing software is build on java thus it is platform independent it can be used with any operating system as the customer use this software he can perform all the basic task with files and can handle different types of file at the same time files having different extentions like .html, .cpp, .xml,.php and many more typically these files are saved as .txt extention as default

This Text Editor is a software that provide a variety of features for the user the user can use this editor at any time and can perform any task without being wear out.

* shortcuts for common tasks like new, open and save.
* Notepad can edit traditional 8-bit text files as well as Unicode text files.
* Notepad accepts text from the Windows clipboard
* Simple text editors like Notepad may be used to edit text with markup, such as HTML
* Notepad Includes the ability to change the font and font style.
* Right Click Feature for Performing Easy task of Text Editing like cut, copy, paste etc.
* When clipboard data with multiple formats is pasted into Notepad, the program will only accept text in the txt format.
* Time and date functional ability can easily insert Time and Date Stamp in a Text Format in a the notepad.

**1.2 AIM/OBJECTIVE**

As we all know, Java is platform independent Language. It can Run on any Operating System which has JVM installed.Notepad of Windows is the most simplest Text Editor Till Now.Simplified and User Friendly Interface of Notepad make it most reliable for any kind of basic Text Editing. One can not use Windows notepad in other Operating Systems for Simplify the use of Notepad of Windows in Other Operating System with ease, The Notepad is Developed in Java.

This way, this popular tool of windows can be run on any Operating System.

**1.3 ADVANTAGE / DISADVANTAGE**

The advantages of creating a website in Notepad is that it is a very simple program. The disadvantage is that you will not be able to see visual feedback. You need to use a more advanced program to do this such as Dream come true.

**2. SYSTEM ANALYSIS**

**2.1 REQUIREMENT ANALYSIS**

The main aim of this phase is to understand the exact requirement of the customer and to document them properly. This phase consists of two distinct activities, namely Requirement gathering and requirement specification.

**Requirements gathering and analysis**: This activity consists of first gathering the requirement and then analyzing the gathered requirement. The goal of the requirements gathering activity is to collect all the relevant information regarding to the product to be developed from the customer with a view to clearly understand the customer requirement. Once the requirements have been gathered , the analysis activity is taken up. The goal of requirement analysis activity is to weed out all the incompleteness and inconsistencies in these requirements.

The qualities we desire of a problem solution other than those concerning its functionality, e.g. its robustness, its efficiency, its security, its extensibility, its maintainability, its portability, etc.

**2.2 NONFUNCTIONAL REQUIREMENTS**

Our “Catalog Management Application” meets these following non functional requirements

**Quality Requirements:**

The application has maintained good search options which generate accurate information.

**Usability**:

Once the application is developed it can be used for different search purposes. And all the users use the system according to their requirements.

**Security:**

The application is available for all users who are registered with us. For this user has to provide accurate details.

**Supportability*:***

The application is more supportable. It allows modifying without causing any high level architecture changes. These changes can be done managerial people.

**Scalability:**

The application is more scalable. It can be use full for different search purposes and can be used for different people.

**2.3 REQUIREMENT DEFINITION AND SPECIFICATION:**

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations. Parameters such as operating speed, response time, availability, portability, maintainability, footprint, security and speed of recovery from adverse events are evaluated.

* 1. **HARDWARE REQUIREMENT**

**Development Side Requirements**

* Processor : Pentium
* Hard Disk: 1GB
* RAM: 1GB
* Monitor : Any standard monitor with true color

**User Side Requirement**

* Processor : Pentium
* Hard Disk: 1 GB
* RAM: 256MB or more
* Monitor: Any standard monitor with true colors
  1. **SOFTWARE REQUIREMENT**

**1.Development Side Requirements**

Operating System : Windows XP

Language : Java

**2.User Side Requirements**

Operating System : Any operating system.

**3. TOOLS AND TECHNOLOGY:**

This project is developed using the following technology:

**Technology- Java:** Now we will explain the technologies which are used in the implementation of this project.

**JAVA:**

Java is an object oriented programming language and it was intended to serve as a new way to manage software complexity. Java refers to a number of computer software products and specification from SUN Microsystems that together provide a system for developing application software and deploying it in a better platform. Java is used in a variety of computing platforms from embedded devices and mobile phones.Java was originally developed by James Gosling at SUN Microsystems and released in 1995 as a core component of SUN Microsystems’s Java platform.

Java quickly became popular with the advent of Java2 (released initially as J2SE). And then J2EE targeted as enterprise application and J2ME as mobile applications.

We are using Java because it is:

**Simple**: -It is simple to use and has less complexity in syntaxes even a learing programmer can use and learn it efficiently.

1. **Object Oriented**: -

Java is a pure object oriented programming language. It basically emphasize on taking object as the centre and working on whole. Java is mainly used for this purpose.

1. **Platform independent**: -

Java is a platform independent language. This means that it simply not dependent on any particular editor. Even the program can be sifted and run at different systems.

1. **Distributed: -**

Java is also distributed in nature. Java can be said as been implemented on distributed systems and is easily implemented to program modularity in Java.

1. **Secure**:-

Java proves nice at security. Even the programs and applets in java don’t accept highly corrupted data.

1. **Interpreted**: -

Java involves the working of both an interpreter and a compiler which makes at good at efficient working.

1. **Multithreaded**: -After the concept of multi programming and multi processing the task**.** Multithreaded was widely in acceptance and java truly support it.

**About Java**

JAVA is an object oriented programming language and it was intended to serve as a new way to manage software complexity. Java refers to a number of computer software products and specifications from Sun Microsystems that together provide a system for developing application software and deploying it in a cross-platform environment. Java is used in a variety of computing platforms from embedded devices and mobile phones on the low end, to enterprise servers and supercomputers on the high end. Java is nearly everywhere in mobile phones, Web servers and enterprise applications, and while less common on desktop computers; Java applets are often used to provide improved functionality while browsing the World Wide Web.

**Some advantages of JAVA:**

* It is an open source, so users do not have to struggle with heavy license fees each year
* Platform independent
* Java API's can easily be accessed by developers
* Java perform supports garbage collection, so memory management is automatic
* Java always allocates objects on the stack
* Java embraced the concept of exception specifications
* Multi-platform support language and support for web-services
* Using JAVA we can develop dynamic web applications.

Another advantage of JAVA is that, ones the program is written in java we can run it anywhere means that application developed through Java is platform independent. JAVA based enterprise applications perform well because stable JAVA standards help developers to create multilevel applications with a component based approach.

**4. PROCESS MODEL ADOPTED**

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic

during testing phase. To develop this project I have used **Spiral Model.**

**SPIRAL MODEL** was defined by Barry Boehm in his 1988 article, “A spiral Model ofSoftware Development and Enhancement. This model was not the first model to discussiterative development, but it was the first model to explain why the iteration models. Asoriginally envisioned, the iterations were typically 6 months to 2 years long. Each phase startswith a design goal and ends with a client reviewing the progress thus far. Analysis andengineering efforts are applied at each phase of the project, with an eye toward the end goal ofthe project.

**SPIRAL MODEL**



**5. DESIGN**

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.

**5.1 E-R DIAGRAM:**

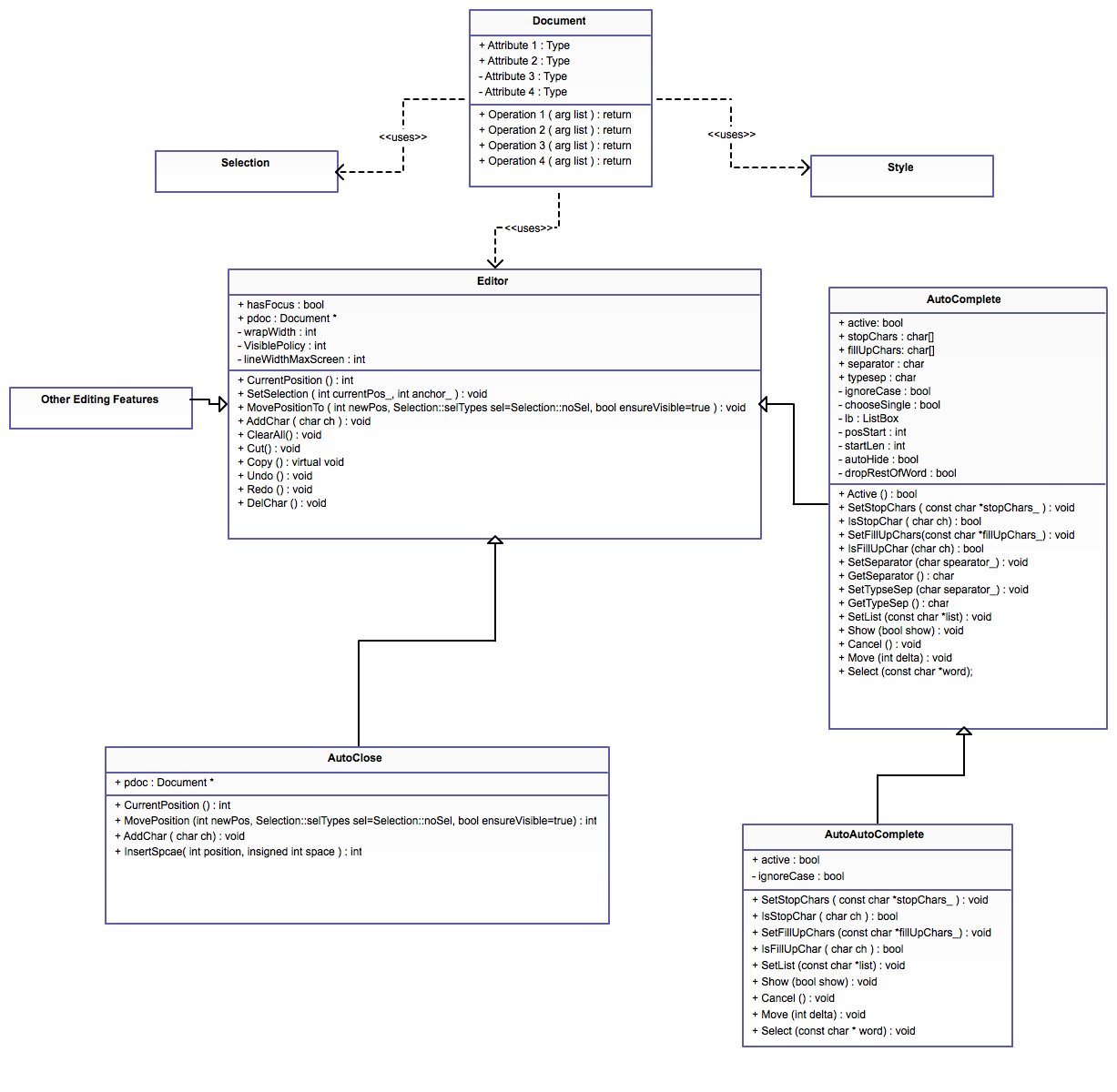
User 1

Editior

User 2

User n

* 1. **CLASS DIAGRAM:**

****

**6. CODING**

**SOURCE CODE**

import javax.swing.ButtonGroup;

import javax.swing.JRadioButtonMenuItem;

import javax.swing.SwingUtilities;

import javax.swing.JColorChooser;

import javax.swing.ImageIcon;

import java.awt.BorderLayout;

import java.awt.Container;

import java.awt.Font;

import java.awt.Color;

import java.awt.GraphicsEnvironment;

import java.awt.Point;

import javax.swing.event.CaretListener;

import javax.swing.event.CaretEvent;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.InputEvent;

import java.awt.event.KeyEvent;

import java.awt.print.PrinterException;

import java.awt.print.PrinterJob;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import javax.swing.JCheckBoxMenuItem;

import javax.swing.JFileChooser;

import javax.swing.JFrame;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

public class notepad extends JFrame implements ActionListener

{

Container c;

public JScrollPane sc;

public JTextArea t;

public JMenuBar status;

public JMenu jm\_status;

private JMenuBar menubar;

private JMenu file;

private JMenuItem file\_new;

private JMenuItem file\_open;

private JSeparator file\_sep1;

private JMenuItem file\_save;

private JMenuItem file\_save\_as;

private JSeparator file\_sep2;

private JMenuItem file\_print;

private JMenuItem file\_close;

private JMenuItem file\_exit;

private JMenu edit;

private JMenuItem edit\_undo;

private JMenuItem edit\_redo;

private JMenuItem edit\_copy;

private JMenuItem edit\_cut;

private JMenuItem edit\_paste;

private JMenuItem edit\_delete;

private JMenuItem edit\_find;

private JMenuItem edit\_find\_next;

private JMenuItem edit\_replace;

private JSeparator edit\_sep3;

private JMenuItem edit\_selectall;

private JMenuItem edit\_timedate;

private JMenu option;

private JMenuItem count;

private JSeparator option\_sep1;

private JMenuItem foreground;

private JMenuItem background;

private JMenu format;

private JMenuItem format\_font;

private JSeparator format\_sep1;

private JMenu convert;

private JMenu lookAndFeelMenu;

private JMenuItem str2uppr, str2lwr;

private JSeparator format\_sep2;

private JCheckBoxMenuItem format\_wordwarp;

private JMenu help;

private JMenuItem help\_detail;

private JMenuItem help\_about;

UndoManager undo = new UndoManager();

find finder;

font\_chooser fc;

String path, content;

public notepad(){

super("Smartpad");

try

{

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

}

catch(Exception ex)

{

ex.printStackTrace();

}

setIconImage((new ImageIcon("notepad.jpg")).getImage());

c = getContentPane();

t = new JTextArea("", 5,5);

t.setFont(new Font("Verdana",Font.PLAIN, 12));

sc = new JScrollPane(t, sc.VERTICAL\_SCROLLBAR\_AS\_NEEDED, sc.HORIZONTAL\_SCROLLBAR\_AS\_NEEDED); //adding scrollbar to text area;

c.add(sc);

status = new JMenuBar();

c.add(status, BorderLayout.SOUTH);

jm\_status=new JMenu();

menubar = new JMenuBar();

file = new JMenu("File");

file\_new = new JMenuItem("New");

file\_new.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_N, InputEvent.CTRL\_MASK));

file\_new.addActionListener(this);

file.add(file\_new);

file\_open = new JMenuItem("Open");

file\_open.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_O, InputEvent.CTRL\_MASK));

file\_open.addActionListener(this);

file.add(file\_open);

file\_sep1 = new JSeparator();

file.add(file\_sep1);

file\_save = new JMenuItem("Save");

file\_save.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_S, InputEvent.CTRL\_MASK));

file\_save.addActionListener(this);

file.add(file\_save);

file\_save\_as = new JMenuItem("Save As");

file\_save\_as.addActionListener(this);

file.add(file\_save\_as);

file\_sep2 = new JSeparator();

file.add(file\_sep2);

file\_print = new JMenuItem("Print");

file\_print.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_P,

edit = new JMenu("Edit");

edit\_undo = new JMenuItem("Undo");

edit\_undo.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_Z,

edit.add(edit\_cut);

edit\_paste = new JMenuItem("Paste");

edit\_paste.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_V, InputEvent.CTRL\_MASK));

edit\_paste.addActionListener(this);

InputEvent.CTRL\_MASK));

str2uppr.addActionListener(this);

convert.add(str2uppr);

str2lwr = new JMenuItem("To Lowercase...");

str2lwr.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_DOWN, InputEvent.CTRL\_MASK));

str2lwr.addActionListener(this);

convert.add(str2lwr);

format.add(convert);

lookAndFeelMenu = new JMenu("Look and Feel");

format.add(lookAndFeelMenu);

format\_sep2 = new JSeparator();

format.add(format\_sep2);

format\_wordwarp = new JCheckBoxMenuItem("Word Warp");

format\_wordwarp.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_W,

menubar.add(format);

option=new JMenu("Option");

count=new JMenuItem("Count");

foreground=new JMenuItem("Foreground");

background=new JMenuItem("Background");

count.addActionListener(this);

foreground.addActionListener(this);

background.addActionListener(this);

count.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_C, InputEvent.ALT\_MASK));

foreground.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_F, InputEvent.ALT\_MASK));

background.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_B, InputEvent.ALT\_MASK));

option.add(count);

option\_sep1 = new JSeparator();

option.add(option\_sep1);

option.add(foreground);

option.add(background);

menubar.add(option);

help = new JMenu("Help");

help\_about = new JMenuItem("About");

help\_about.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_F1, 0));

help\_detail=new JMenuItem("Detail");

help\_detail.addActionListener(this);

help\_about.addActionListener(this);

help.add(help\_about);

help.add(help\_detail);

menubar.add(help);

// find\_window

finder = new find(this);

finder.setDefaultCloseOperation(JFrame.HIDE\_ON\_CLOSE);

// font chooser

fc = new font\_chooser// set window size

int w = 600;

int h = 700;

setSize(w, h);

// set window position

Point center = GraphicsEnvironment.getLocalGraphicsEnvironment().getCenterPoint();

setLocation(center.x-w/2, center.y-h/2);

setVisible(true);

path = "";

manageLookAndFeels();

updateStatus(1,1);

t.addCaretListener(new CaretListener() {

public void caretUpdate(CaretEvent e) {

JTextArea editArea = (JTextArea)e.getSource(); //EventObject

int linenum = 1;

int columnnum = 1;

try {

int caretpos = editArea.getCaretPosition(); //JTextComponent

linenum = editArea.getLineOfOffset(caretpos);

columnnum = caretpos - editArea.getLineStartOffset(linenum)+1;

linenum += 1;

}

catch(Exception ex) { }

updateStatus(linenum, columnnum);

}

});

}

public void updateStatus(int linenumber, int columnnumbe

String str="Line: " + linenumber + " , Column: " + columnnumber;

jm\_status.setText(str);

status.add(jm\_status);

}

public void actionPerformed(ActionEvent e){

if(e.getSource()==file\_new)

file\_new();

else if(e.getSource()==file\_open)

file\_open();

else if(e.getSource()==file\_save)

file\_save();

else if(e.getSource()==file\_save\_as)

file\_save\_as();

else if(e.getSource()==file\_print)

file\_print();

else if(e.getSource()==file\_close)

file\_close();

else if(e.getSource()==file\_exit)

file\_exit();

setTitle("Smartpad");

}

else

{

int a = JOptionPane.showConfirmDialog(null, "The text has been changed\nDo you want to save the changes?");

if(a==0)

file\_save();

else if(a==1)

{

t.setText("");

path = "";

setTitle("Smartpad");

}

else if(a==2)

return;

}

}

public void file\_open(){

JFileChooser fc = new JFileChooser();

fc.setFileSelectionMode(JFileChooser.FILES\_ONLY);

int r=fc.showOpenDialog(this);

if(r==fc.CANCEL\_OPTION)

return;

File myfile = fc.getSelectedFile();

if(myfile == null || myfile.getName().equals(""))

{

JOptionPane.showMessageDialog(this, "Select a file!", "Error", JOptionPane.ERROR\_MESSAGE);

return;

}

try

{

BufferedReader input = new BufferedReader(new FileReader(myfile));

StringBuffer str = new StringBuffer();

String line;

while((line = input.readLine()) != null) // st is declared as string above

str.append(line+"\n");

t.setText(str.toString());

content = t.getText();

path = myfile.toString();

setTitle(myfile.getName()+" - Smartpad");

}

catch(FileNotFoundException e)

{

JOptionPane.showMessageDialog(null, "File not found: "+e);

}

catch(IOException e)

{

JOptionPane.showMessageDialog(null, "IO ERROR: "+e);

}

}

public void file\_save(){

if(path.equals(""))

{

file\_save\_as();

return;

}

try

{

FileWriter fw = new FileWriter(path);

fw.write(t.getText());

content = t.getText();

fw.close();

}

catch(IOException i)

{

JOptionPane.showMessageDialog(this,"Failed to save the file","Error",JOptionPane.ERROR\_MESSAGE);

}

}

public void file\_save\_as(){

JFileChooser fc = new JFileChooser();

fc.setFileSelectionMode(JFileChooser.FILES\_ONLY);

int r = fc.showSaveDialog(this);

if(r==fc.CANCEL\_OPTION)

return;

File myfile = fc.getSelectedFile();

if(myfile==null || myfile.getName().equals(""))

{

JOptionPane.showMessageDialog(this,"Please enter a file name!","Error",JOptionPane.ERROR\_MESSAGE);

return;

}

if(myfile.exists())

{

r = JOptionPane.showConfirmDialog(this, "A file with same name already exists!\nAre you sure want to overwrite?");

if(r != 0)

return;

}

try

{

FileWriter fw = new FileWriter(myfile);

fw.write(t.getText());

content = t.getText();

setTitle(myfile.getName()+" - Smartpad");

fw.close();

}

catch(IOException e)

{

JOptionPane.showMessageDialog(this,"Failed to save the file","Error",JOptionPane.ERROR\_MESSAGE);

}

}

public void file\_print() {

PrinterJob printer = PrinterJob.getPrinterJob();

//printer.setPrintable( this);

HashPrintRequestAttributeSet printAttr = new HashPrintRequestAttributeSet();

if(printer.printDialog(printAttr)) // Display print dialog

{ // If true is returned...

try

{

printer.print(printAttr); // then print

}

catch(PrinterException e)

{

JOptionPane.showMessageDialog(this,"Failed to print the file: "+e,"Error",JOptionPane.ERROR\_MESSAGE);

}

}

}

public void file\_close(){

if(t.getText().equals("") || t.getText().equals(content))

{

t.setText("");

path = "";

setTitle("Smartpad");

}

else

{

int a = JOptionPane.showConfirmDialog(null, "The text has been changed\nDo you want to save the changes?");

if(a==0)

file\_save();

else if(a==1)

{

t.setText("");

path = "";

setTitle("Smartpad");

}

else if(a==2)

return;

}

}

public void file\_exit(){

if(t.getText().equals("") || t.getText().equals(content))

System.exit(0);

else

{

int b = JOptionPane.showConfirmDialog(null, "The text has been changed.\nDo you want to save the changes?");

if(b==0)

file\_save();

else if(b==1)

System.exit(0);

else if(b==2)

return;

}

}

public void edit\_undo() {

if( undo.canUndo())

{

try

{

undo.undo();

}

catch(CannotUndoException e)

{

}

}

}

public void edit\_redo(){

if( undo.canRedo())

{

try

{

undo.redo();

}

catch(CannotRedoException e)

{

}

}

}

public void edit\_cut(){

t.cut();

}

public void edit\_copy(){

t.copy();

}

public void edit\_paste(){

t.paste();

}

public void edit\_delete(){

String temp = t.getText();

t.setText(temp.substring(0, t.getSelectionStart())+temp.substring(t.getSelectionEnd()));

}

public void edit\_find(){

finder.setVisible(true);

}

public void str2uppr(){

try

{

int start = t.getSelectionStart();

int end = t.getSelectionEnd();

String temp1 = t.getText().substring(0,start);

String temp2 = t.getText().substring(end);

String conv = t.getSelectedText().toUpperCase();

t.setText(temp1+conv+temp2);

t.select(start, end);

}

catch(NullPointerException e){}

}

public void str2lwr(){

try

{

int start = t.getSelectionStart();

int end = t.getSelectionEnd();

String temp1 = t.getText().substring(0,start);

String temp2 = t.getText().substring(end);

String conv = t.getSelectedText().toLowerCase();

t.setText(temp1+conv+temp2);

t.select(start, end);

}

catch(NullPointerException e){}

}

public void format\_wordwarp(){

if(t.getLineWrap()==false)

t.setLineWrap(true);

else

t.setLineWrap(false);

if(format\_wordwarp.getState())

status.setVisible(false);

else

status.setVisible(true);

}

public void help\_about(){

String dtl ="Created By :" +

"\N RUCHI SAXENA" +

"\nComputer Science & Engineering" +

"\NR.K.D.F. INSTITUTE OF SCIENCE & TECHNOLOGY"+

"\n\nContact us at:" + "\ns.saxenaruchi@gmail.com"+

"\nBuilt Date: April 6, 2014";

JOptionPane.showMessageDialog(t,dtl,"About Smartpad",JOptionPane.PLAIN\_MESSAGE,new ImageIcon("notepad.jpg"));

}

public void help\_detail()

{

String dtl="Smartpad is a basic text editor that you can \n"+

"use to create simple documents.The most \n"+

"common use of Smartpad is to view or edit \n"+

"text(.txt) files,but many users find Smartpad\n"+

"a simple tool for creating Web pages." ;

JOptionPane.showMessageDialog(t,dtl,"Detail Smartpad",JOptionPane.INFORMATION\_MESSAGE);

}

public void counting()

{

String str=t.getText();

StringTokenizer st=new StringTokenizer(str," \n");

int words=st.countTokens();

st=new StringTokenizer(str,"\n");

int lines=st.countTokens();

int characters=str.length();

JOptionPane.showMessageDialog(t,"characters:"+characters+"\nwords:"+words+"\nlines:"+lines,"counting",JOptionPane.INFORMATION\_MESSAGE);

}

public void manageLookAndFeels()

{

final LookAndFeelInfo lookAndFeelInfo[]=UIManager.getInstalledLookAndFeels();

ButtonGroup lookAndFeelGroup=new ButtonGroup();

for(int i=0;i<lookAndFeelInfo.length;i++)

{

String lookAndFeelName=lookAndFeelInfo[i].getName();

JRadioButtonMenuItem radioButtonMenuItem=new JRadioButtonMenuItem(lookAndFeelName);

final int j=i;

final JFrame frame=this;

lookAndFeelMenu.add(radioButtonMenuItem);

lookAndFeelGroup.add(radioButtonMenuItem);

radioButtonMenuItem.addActionListener(new ActionListener() { }

});

if(lookAndFeelName.equalsIgnoreCase("Metal"))

{

radioButtonMenuItem.setSelected(true);

}

}

}

import java.awt.Checkbox;

import java.awt.GraphicsEnvironment;

import java.awt.Label;

import java.awt.Point;

import java.awt.TextField;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

public class find extends JFrame implements ActionListener

{

int startIndex=0;

Label l1, l2;

TextField tf, tr;

JButton find\_btn, find\_next, replace, replace\_all, cancel;

notepad samp;

public find(notepad mynote)

{

super("Find / Replace");

samp = mynote;

l1 = new Label("Find What: ");

l2 = new Label("Replace With: ");

tf = new TextField(30);

tr = new TextField(30);

find\_btn = new JButton("Find");

find\_next = new JButton("Find Next");

replace = new JButton("Replace");

replace\_all = new JButton("Replace All");

cancel = new JButton("Cancel");

setLayout(null);

int label\_w = 80;

int label\_h = 20;

int tf\_w = 120;

l1.setBounds(10,10, label\_w, label\_h);

add(l1);

tf.setBounds(10+label\_w, 10, tf\_w, 20);

add(tf);

l2.setBounds(10, 10+label\_h+10, label\_w, label\_h);

add(l2);

tr.setBounds(10+label\_w, 10+label\_h+10, tf\_w, 20);

add(tr);

int w = 340;

int h = 150;

setSize(w,h);

// set window position

Point center = GraphicsEnvironment.getLocalGraphicsEnvironment().getCenterPoint();

setLocation(center.x-w/2, center.y-h/2);

setVisible(false);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==find\_btn)

{

find();

}

else if(e.getSource() == find\_next)

{

find\_next();

}

else if(e.getSource() == replace)

{

replace();

}

else if(e.getSource() == replace\_all)

{

replace\_all();

}

else if(e.getSource() == cancel)

{

this.setVisible(false);

}

}

public void find()

{

int select\_start = samp.t.getText().indexOf(tf.getText());

if(select\_start == -1)

{

startIndex = 0;

JOptionPane.showMessageDialog(null, "Could not find "+tf.getText()+"!");

return;

}

if(select\_start == samp.t.getText().lastIndexOf(tf.getText()))

{

startIndex = 0;

}

int select\_end = select\_start+tf.getText().length();

samp.t.select(select\_start, select\_end);

}

public void find\_next()

{

String selection = samp.t.getSelectedText();

try

{

selection.equals("");

}

catch(NullPointerException e)

{

selection = tf.getText();

try

{

selection.equals("");

}

catch(NullPointerException e2)

{

selection = JOptionPane.showInputDialog("Find:");

tf.setText(selection);

}

}

try

{

int select\_start = samp.t.getText().indexOf(selection, startIndex);

int select\_end = select\_start+selection.length();

samp.t.select(select\_start, select\_end);

startIndex = select\_end+1;

if(select\_start == samp.t.getText().lastIndexOf(selection))

{

startIndex = 0;

}

}

catch(NullPointerException e)

{}

}

import java.awt.Color;

import java.awt.Font;

import java.awt.GraphicsEnvironment;

import java.awt.Point;

import java.awt.ScrollPane;

public class font\_chooser implements ActionListener, ListSelectionListener{

static JFrame window = new JFrame("Font Chooser");

notepad samp;

JLabel flist\_label, fsize\_label, fstyle\_label, fprev\_label, preview;

JList flist, fsize, fstyle;

ScrollPane flist\_sc, fstyle\_sc, fsize\_sc;

JButton ok, cancel;

GraphicsEnvironment ge; // graphics env

String font\_names[]; // font names array

Font sample;

String font\_name;

int font\_size, font\_style;

public font\_chooser(notepad ref)

{

samp = ref;

window.setLayout(null);

// creating font name list

ge = GraphicsEnvironment.getLocalGraphicsEnvironment();

font\_names = ge.getAvailableFontFamilyNames();

// font style box

String styles[] = {"Regular", "Bold", "Italic", "Bold Italic"};

fstyle = new JList(styles);

fstyle.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

fstyle\_label = new JLabel("Style: ");

window.add(fstyle\_label);

fstyle\_label.setBounds(140, 10, 80, 20);

fstyle\_sc = new ScrollPane();

fstyle\_sc.add(fstyle);

fstyle\_sc.setBounds(140, 30, 80, 70);

window.add(fstyle\_sc);

fstyle.addListSelectionListener(this);

// font size box

Vector<String> a = new Vector<String>(40, 1);

for (int i = 8; i <= 100; i += 2)

a.addElement(String.valueOf(i));

fsize = new JList(a);

fsize.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

fsize\_label = new JLabel("Size: ");

fsize\_sc.add(fsize);

fsize\_sc.setBounds(140, 130, 80, 100);

window.add(fsize\_sc);

fsize.addListSelectionListener(this);

// ok and cancel button

ok = new JButton("OK");

ok.setBounds(230, 30, 75, 20);

ok.addActionListener(this);

window.add(ok);

cancel = new JButton("Cancel");

cancel.setBounds(230, 50, 75, 20);

cancel.addActionListener(this);

window.add(cancel);

fprev\_label = new JLabel("Preview: ");

fprev\_label.setBounds(10, 230, 300, 20);

window.add(fprev\_label);

int w = 320;

int h = 380;

window.setSize(w,h);

// set window position

Point center = GraphicsEnvironment.getLocalGraphicsEnvironment().getCenterPoint();

window.setLocation(center.x-w/2, center.y-h/2+25);

window.setDefaultCloseOperation(JFrame.HIDE\_ON\_CLOSE);

window.setVisible(false);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==ok)

ok();

else if(e.getSource()==cancel)

cancel();

}

public void valueChanged(ListSelectionEvent l)

{

if(l.getSource()==flist)

private void changeFontSample()

{

try

{

font\_name = flist.getSelectedValue().toString();

}

catch(NullPointerException npe)

{

font\_name = "Verdana";

}

try

{

font\_style = getStyle();

}

catch(NullPointerException npe)

{

font\_style = Font.PLAIN;

} sample = new Font(font\_name, font\_style, font\_size);

preview.setFont(sample);

}

private int getStyle()

{

if( fstyle.getSelectedValue().toString().equals("Bold") )

return Font.BOLD;

if(fstyle.getSelectedValue().toString().equals("Italic") )

return Font.ITALIC;

if(fstyle.getSelectedValue().toString().equals("Bold Italic"))

return Font.BOLD+Font.ITALIC;

return Font.PLAIN;

}

private void ok()

{

try

{

samp.t.setFont(sample);

}

catch(NullPointerException npe){}

this.window.setVisible(false);

}

private void cancel()

{

this.window.setVisible(false);

}

}

import javax.swing.JFrame;

public class MainClass {

public static void main(String args[]){

final notepad mynote = new notepad();

mynote.setDefaultCloseOperation(javax.swing.JFrame.DO\_NOTHING\_ON\_CLOSE);

mynote.addWindowListener(new java.awt.event.WindowAdapter() {

public void windowClosing(java.awt.event.WindowEvent evt) {

mynote.file\_exit();

**7.TESTING AND DEBUGING**

**TESTING TECHNIQUES & STRATEGIES**

It involves the identification of bug/error/defect in the software without correcting it. Normally professionals with a Quality Assurance background are involved in the identification of bugs. Testing is performed in the testing phase.

## Black Box Testing

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

## White Box Testing

White box testing is the detailed investigation of internal logic and structure of the code. White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

## Grey Box Testing

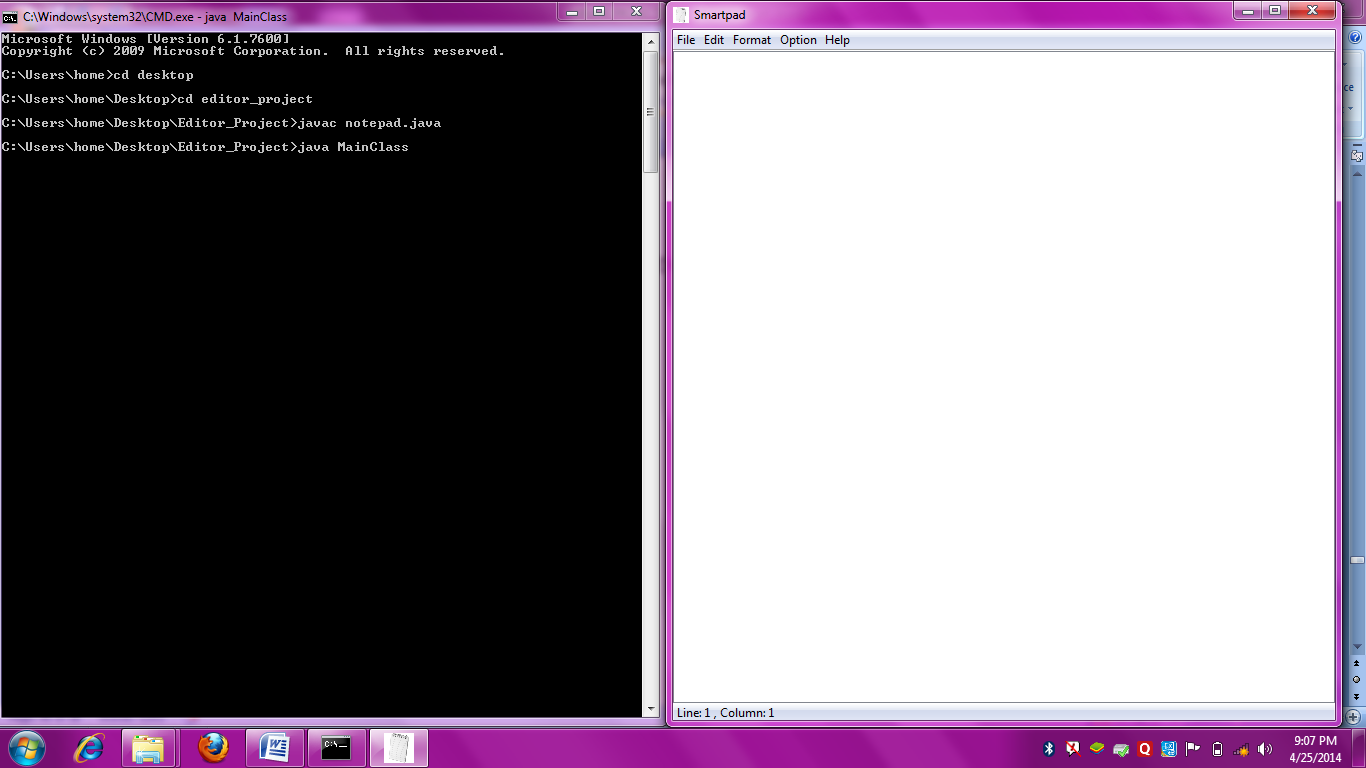
Grey Box testing is a technique to test the application with limited knowledge of the internal workings of an application. In software testing, the term *the more you know the better* carries a lot of weight when testing an application.

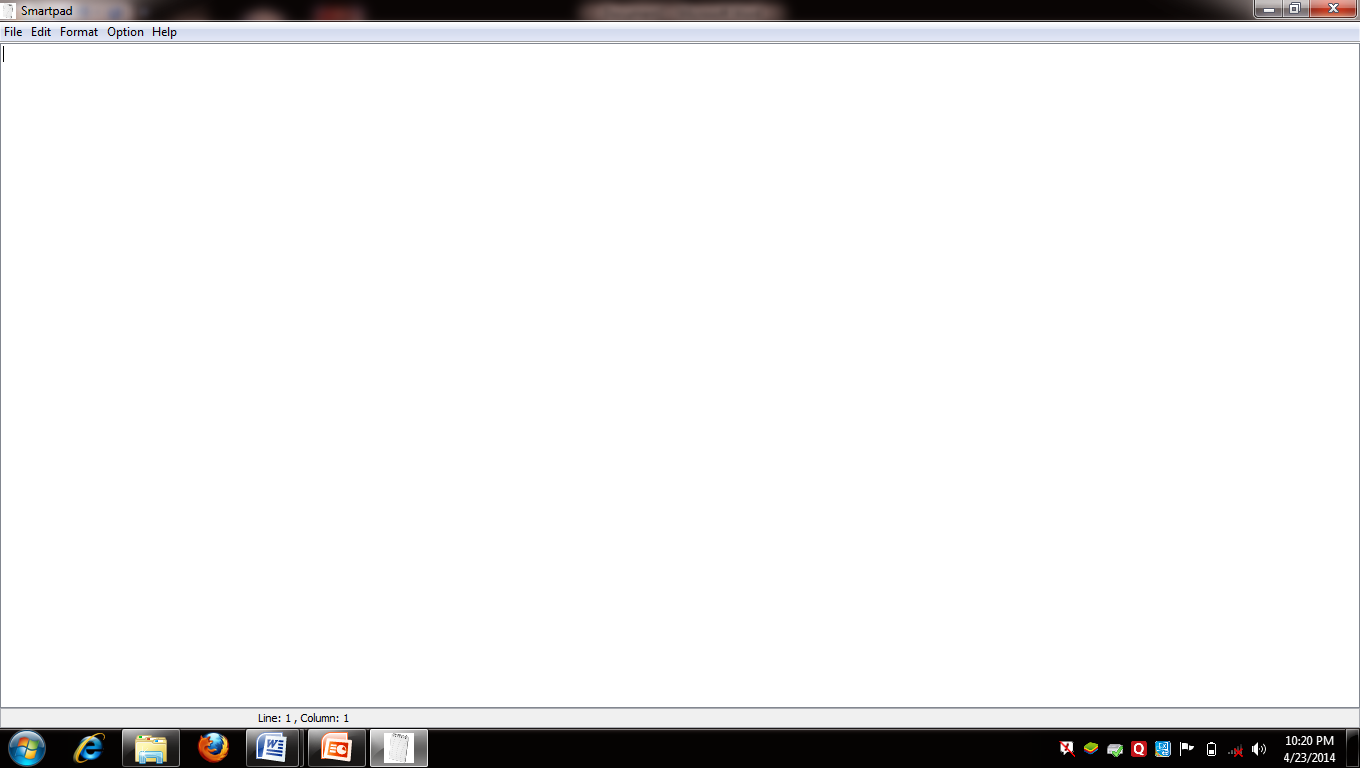
Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black box testing, where the tester only tests the application's user interface, in grey box testing, the tester has access to design documents and the database. Having this knowledge, the tester is able to better prepare test data and test scenarios when making the test plan.

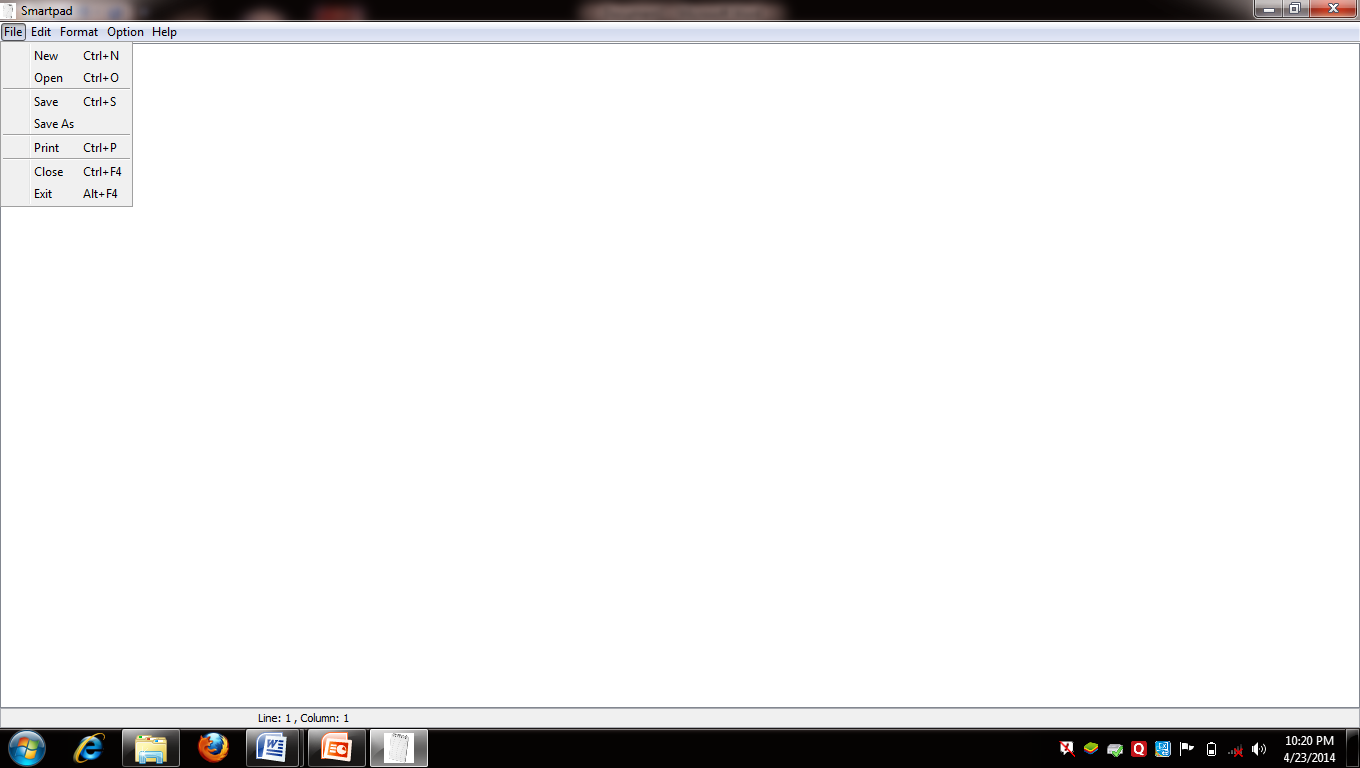
DEBUGGING:It involves identifying, isolating and fixing the problems/bug. Developers who code the software conduct debugging upon encountering an error in the code. Debugging is the part of White box or Unit Testing. Debugging can be performed in the development phase while conducting Unit Testing or in phases while fixing the reported bugging.

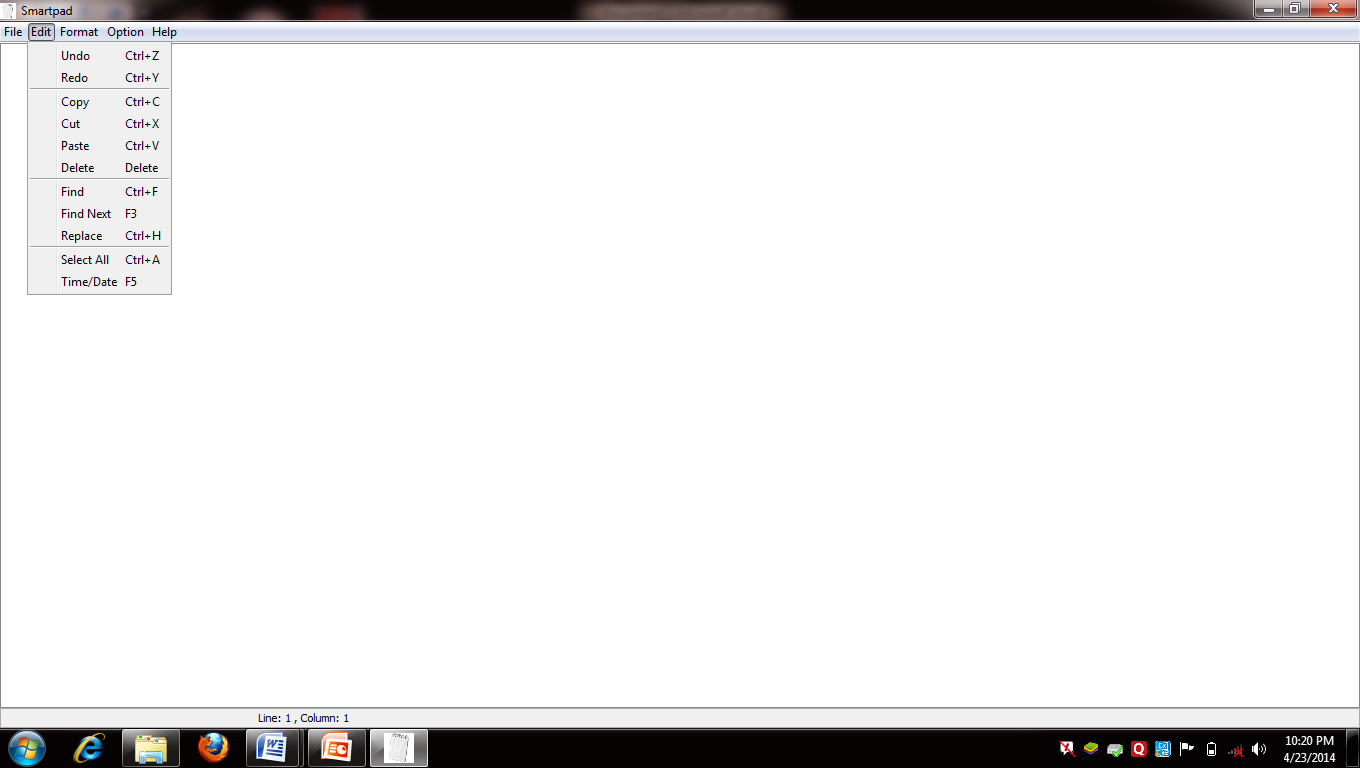
**8. SCREEN SHOTS**

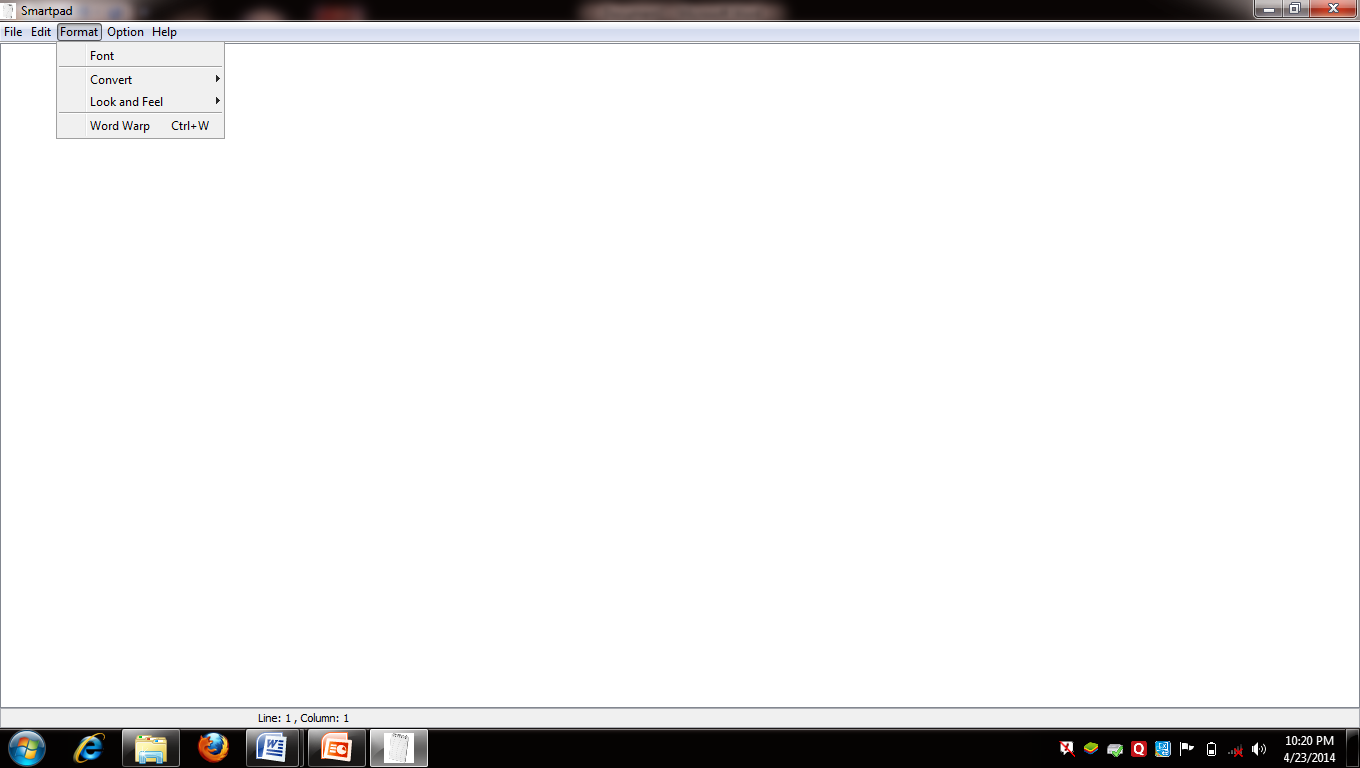
* 1. **FEATURES**
* **STARTUP PAGE :**

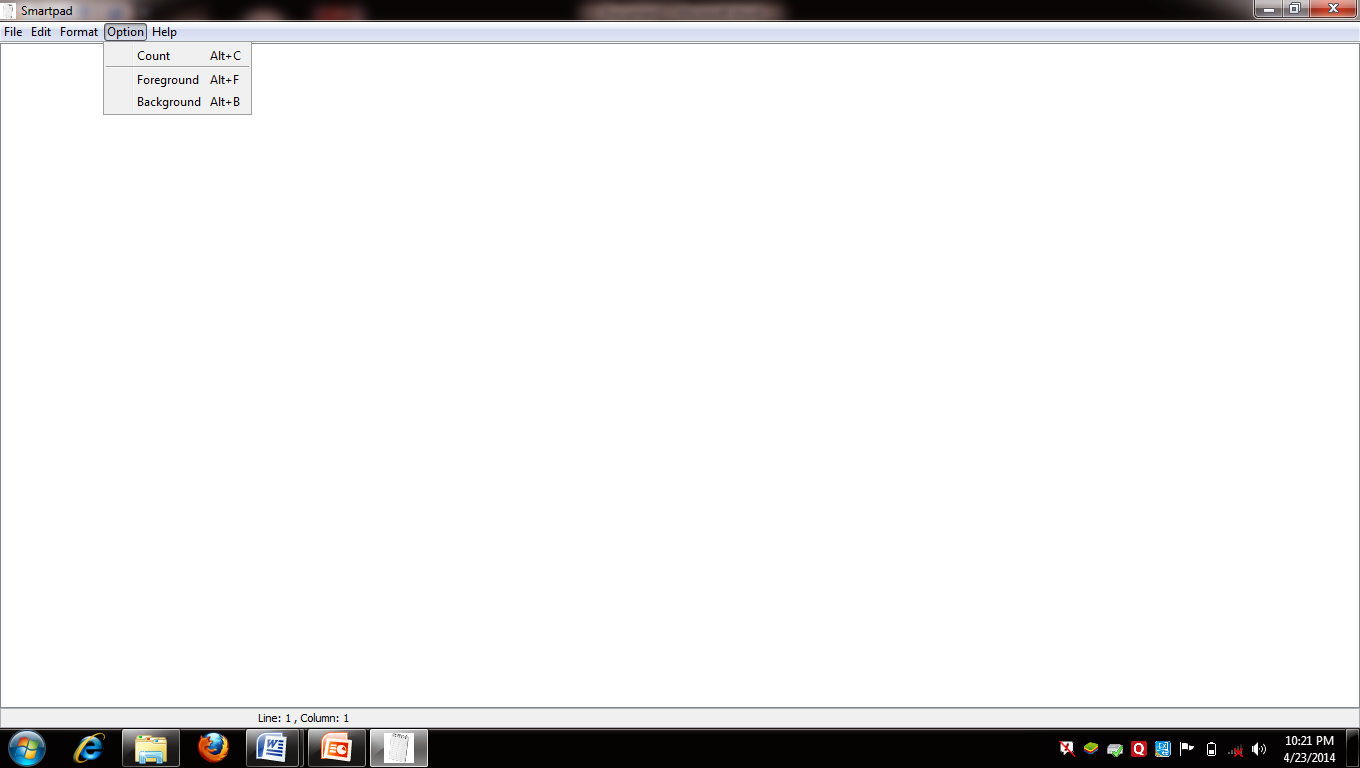
****

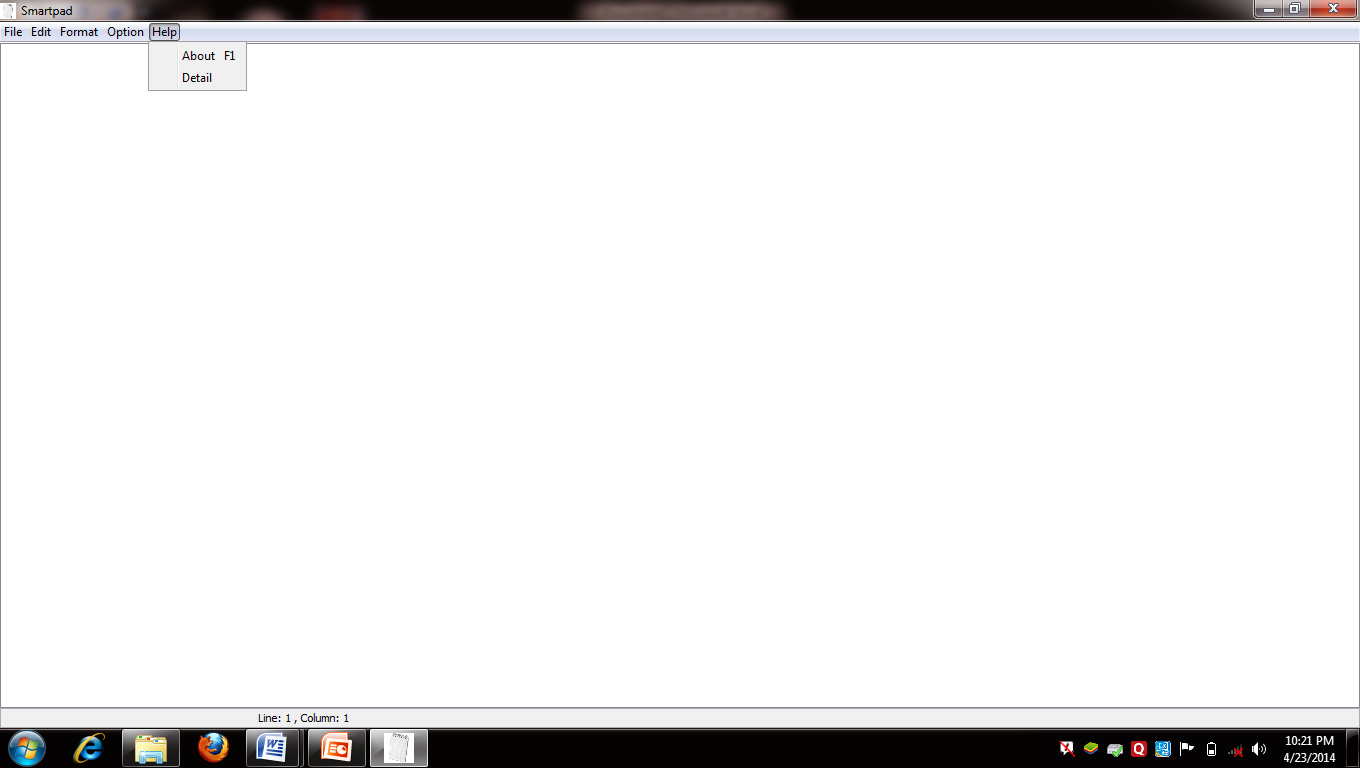
******

****

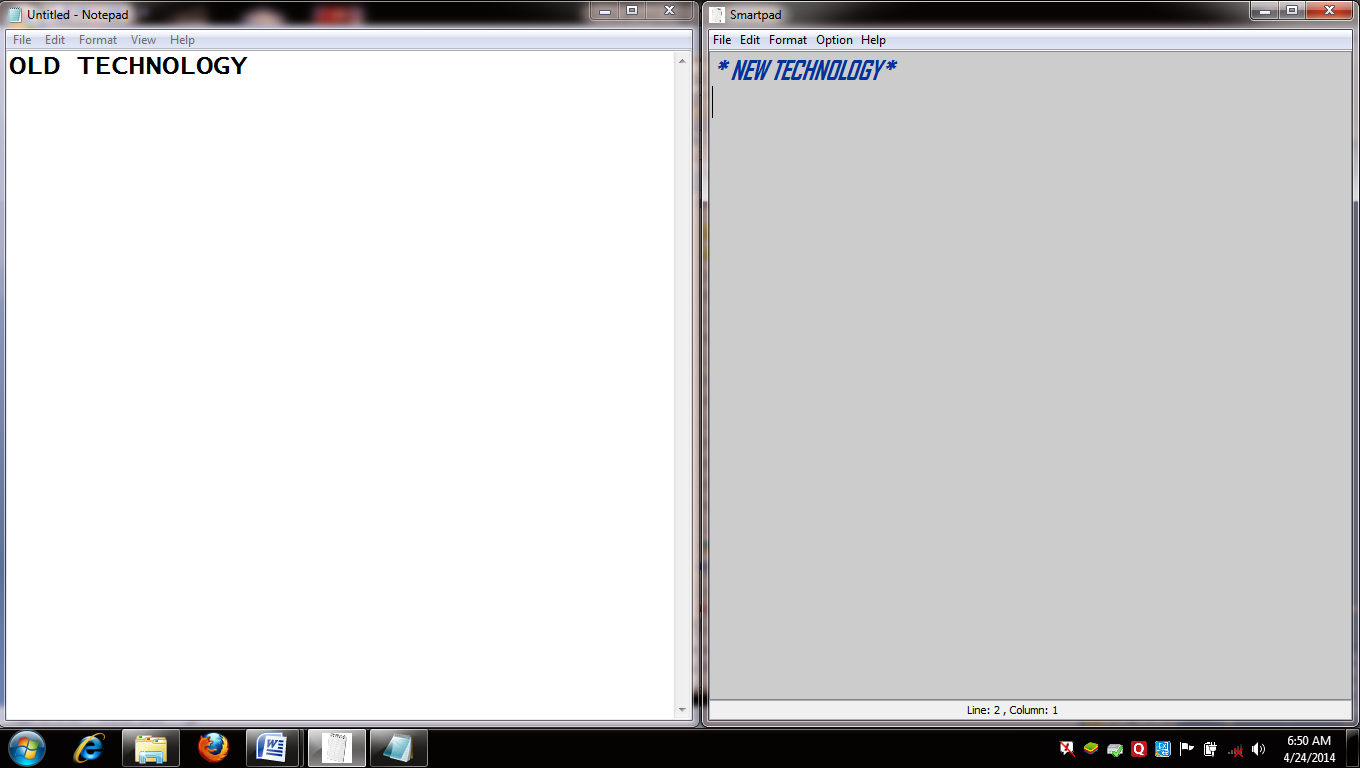
****

****

****

****

**8.2 OLD TECHNOLOGY v/s NEW TECHNOLOGY:**

****

**9. FUTURE SCOPE OF THE PROPOSED SYSTEM**

Present system carries certain drawbacks and limitations as listed below. Current system provides not only includes previous basic function’s but also contain some extra features as per user requirements. As the technology is developing day by day software performance is also improving user wants that for creating types of files he dosen’t require multiple editing software but the work of all these different softwares is done by a single editor which include all the funtionality as required by the customer or the user. As this editing software is build upon java this software can be used upon any operating system as java is a platform independent language.

**10. REFERENCES**

* 1. **BIBLIOGRAPHY**

1. Java- The Complete Reference
2. Software Engineering-A Practioner’s Approach, Roger.s.Pressman

**10.2 WEBSITES**

1. [www.sourcecodeworld.com](http://www.sourcecodeworld.com)
2. [www.codeprojet.com](http://www.codeprojet.com)
3. [www.roseindia.com](http://www.roseindia.com)
4. [www.javaforbegineers.com](http://www.javaforbegineers.com)