

**VISHWAKARMA INSTITUTE OF TECHNOLOGY, PUNE – 411037**

666, Upper Indiranagar , Bibwewadi, Pune 411037

B.Tech Second Year 2020-21

Semester 2

**Subject :** Object Oriented Programming **Code** : **CS2218**

**Home Assignment**

**Type – Design**

**Topic - Text editor using Swing**

**Div.: ME-D Batch 2 – Group 2**

Students Names :

1. D26 Aayush Surawar 11910859

2. D27 Samarth Takbhate 11910739

3. D30 Gopal Tapadiya 11911318

4. D31 Shivam tapre 11910774

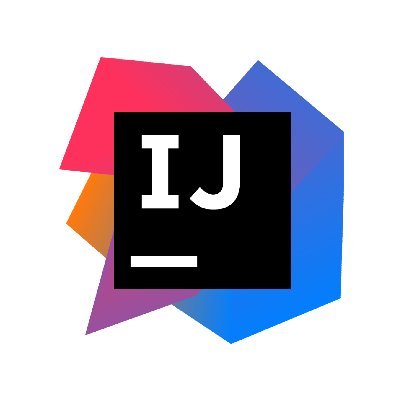
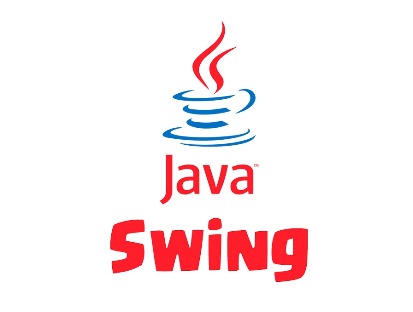
5. D43 Umesh Wanare 11911112

**Guided by – Prof. Dnyaneshwar Kanade.**

**Problem Statement :** To create a simple text editor in Java Swing using a JTextArea, a JMenuBar and add JMenu to it and JMenuItems. All the menu items have actionListener to detect any action.

**Tools :**

* Intellij Idea compiler
* Java swing

**Theory about design -**

**Java swing** :

1. Java Swing is a part of Java Foundation Classes (JFC) that is used to create window-based applications.
2. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.
3. Unlike AWT, Java Swing provides platform-independent and lightweight components.
4. The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

**Java textarea:**

1. JTextArea is a part of java Swing package .
2. It represents a multi line area that displays text. It is used to edit the text .
3. JTextArea inherits JComponent class. The text in JTextArea can be set to different available fonts and can be appended to new text .
4. A text area can be customized to the need of user .

**Java menubar:**

1. JMenuBar is an implementation of menu bar .
2. The JMenuBar contains one or more JMenu objects, when the JMenu objects are selected they display a popup showing one or more JMenuItems .
3. JMenu basically represents a menu . It contains several JMenuItem Object.
4. It may also contain JMenu Objects (or submenu).

**Java swing ActionListener() :**

1. The class which processes the ActionEvent should implement this interface.
2. The object of that class must be registered with a component. The object can be registered using the **addActionListener()** method.
3. When the action event occurs, that object's actionPerformed method is invoked.

**Java JTextPane :**

1. JTextPane is a subclass of JEditorPane class.
2. JTextPane is used for styled document with embedded images and components.
3. It is text component that can be marked up with attributes that are represented graphically.
4. JTextPane uses a DefaultStyledDocument as its default model.

**Java panel :**

1. JPanel, a part of Java Swing package, is a container that can store a group of components.
2. The main task of JPanel is to organize components, various layouts can be set in JPanel which provide better organisation of components, however it does not have a title bar.

Following are the features added to the text editor with their functions:

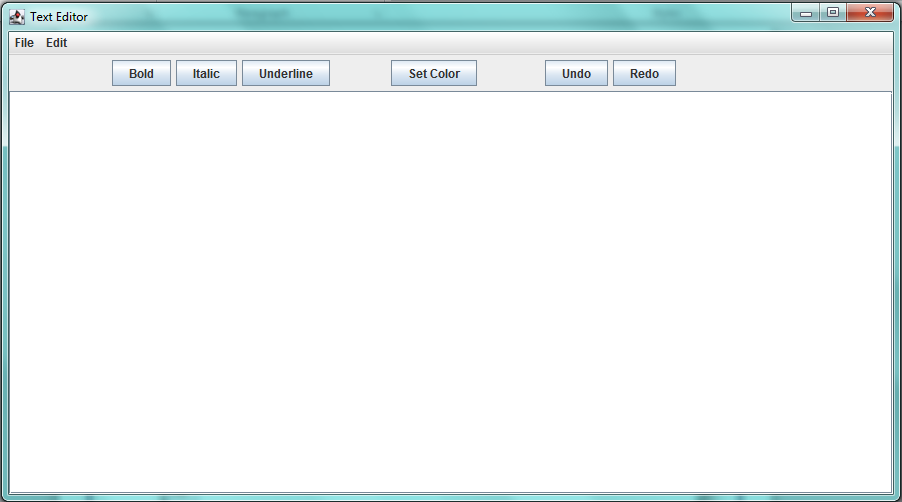


Fig.1 :The Frame

1. The menu added to the menubars-
2. File : It includes 5 menuitems –

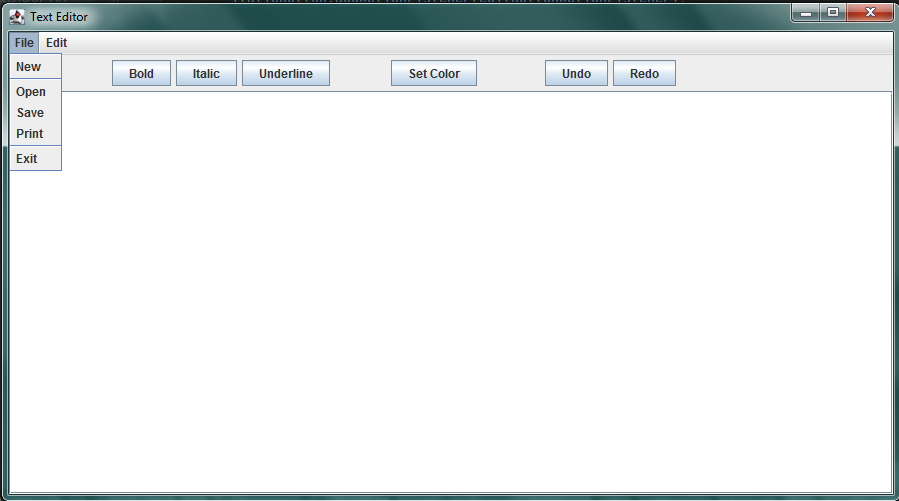


Fig.2 : File menu

1. **New** : This menuitem is used to create a new blank file.
2. **Open** : This menuitem is used to open a file.

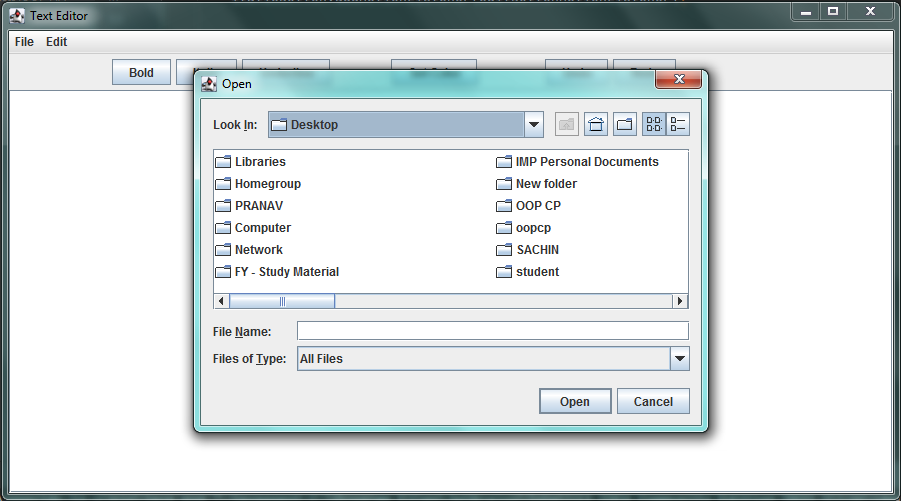


Fig.3 : Open

1. **Save** : This menuitem is used to save a file.

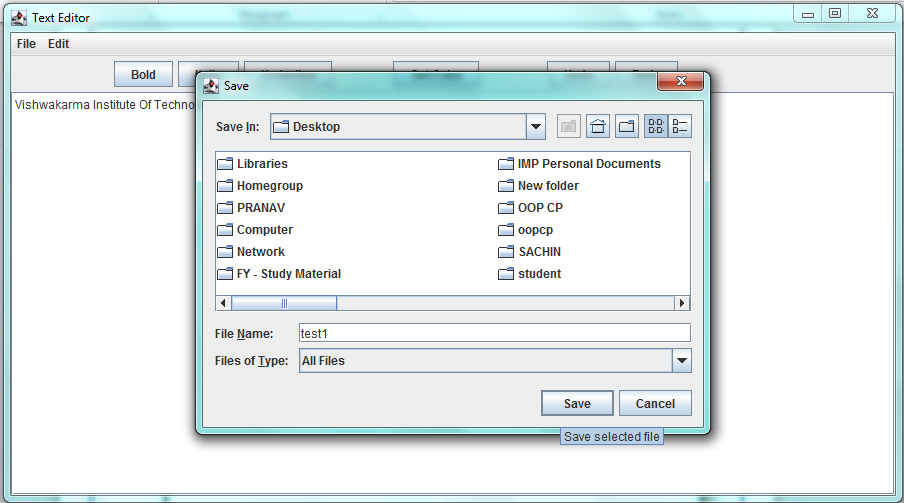


Fig.4 : Save

1. **Print** : This menuitem is used to print the components of the text area.

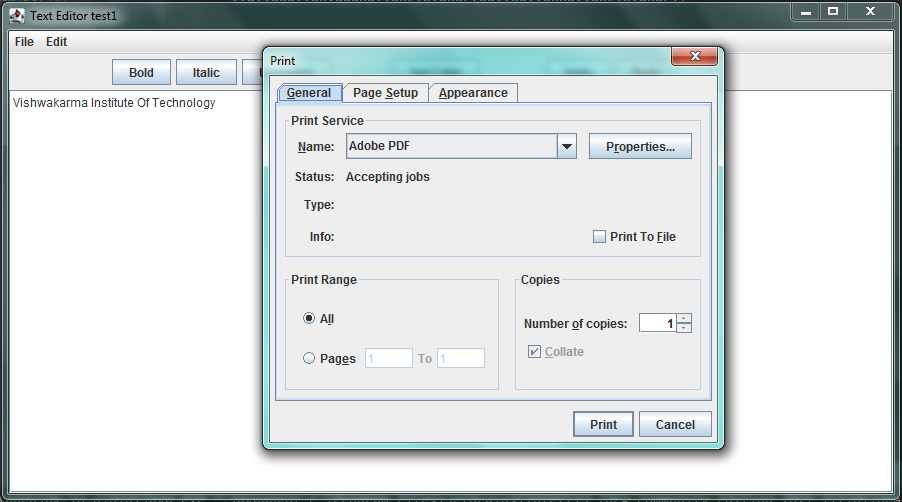


Fig.5 : Print

1. **Exit** : This menuitem closes the frame.
2. Edit : It includes 3 menuitems –

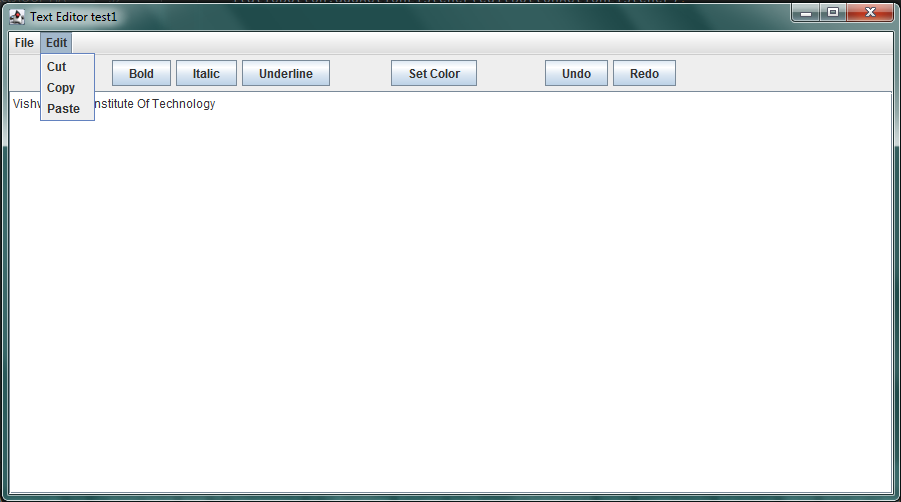


Fig.6 : Edit menu

1. **Cut** : This menuitem is to cut the selected area and copy it to clipboard.
2. **Copy** : This menuitem is to copy the selected area to the clipboard.
3. **Paste** : This menuitem is to paste the text from the clipboard to the text area.
4. The buttons added to panel –
5. **Bold :**

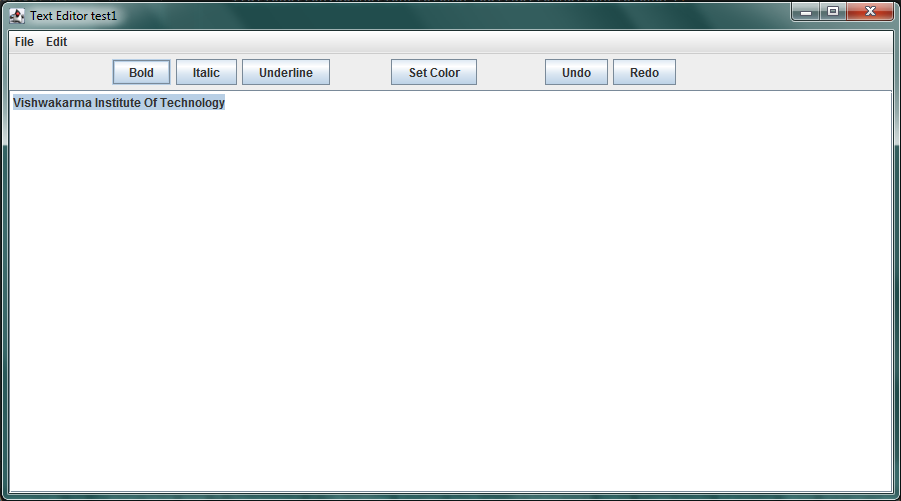
****

Fig.7 : Bold

1. **Italic :**

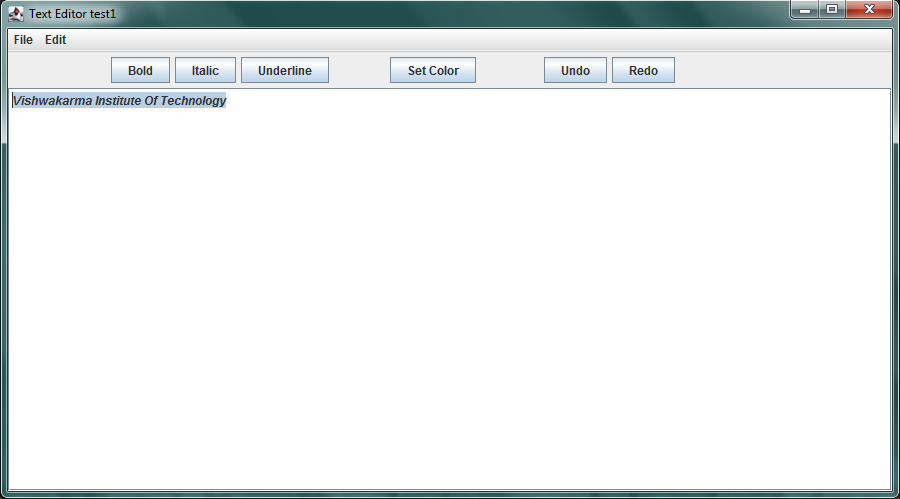
****

Fig.8 : Italic

1. **Underline :**

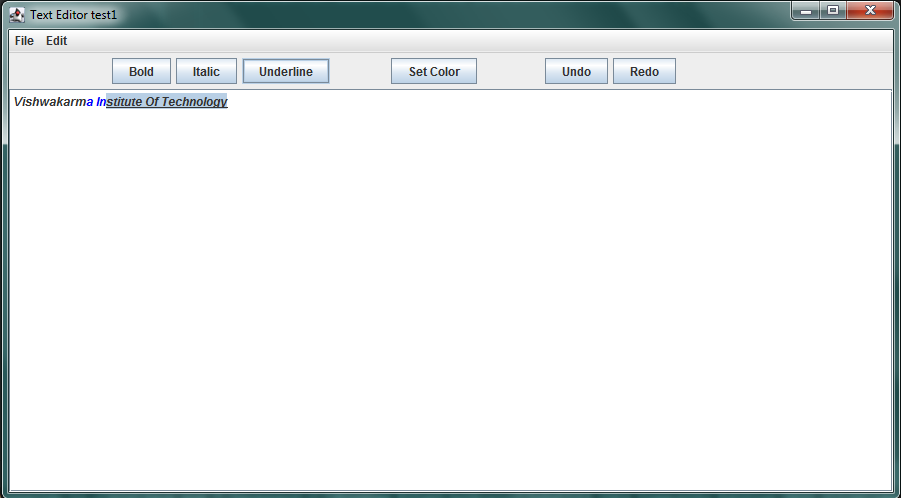
****

Fig.9 : Underline

1. **Set Color:**

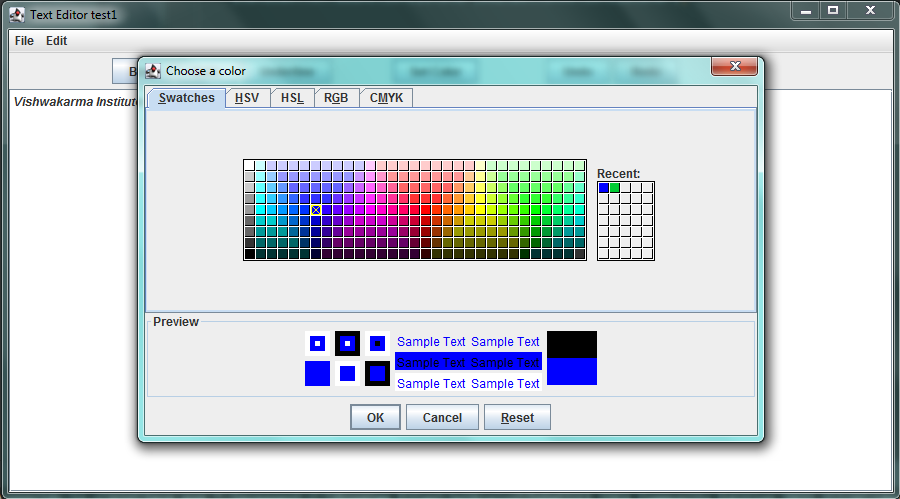
****

Fig.10 : Set Color

1. **Undo:** To undo the action.
2. **Redo:** To redo the action.

Screenshot of frame where all features are applied…

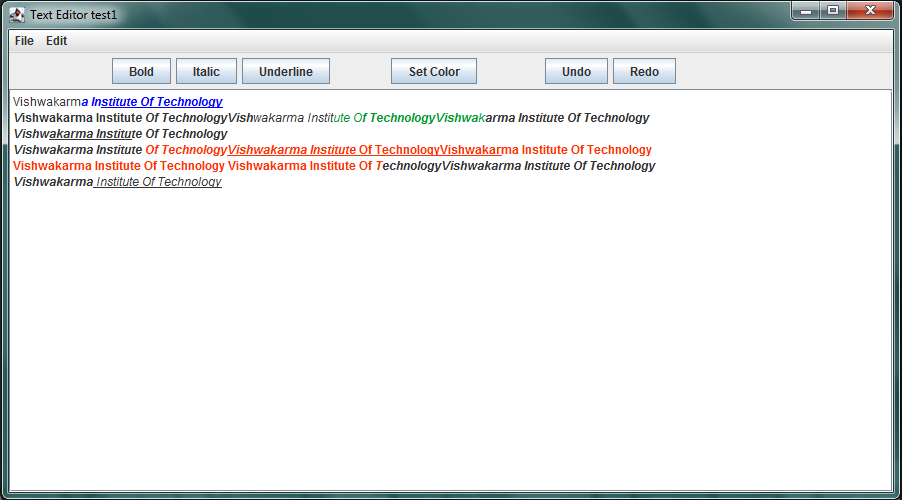
****

Fig.11 : Final frame

**Conclusion** :

1. We have designed a text editor using java swing .
2. It contains the features like open the file , save the file , print the file and open a new file . The cut , copy and paste option are also there to edit text.
3. Additionally the Bold, Italic , Underline , Set color , undo and redo buttons are added to the panel to decorate the text with easy access.

**Future Scope :**

1. As we can see in these design , bold , Italic , underline & set color buttons are added ; so in future we can add the button to change the font of the text also we can add a feature to change the font size of text .
2. There is chance to add feature like alignment of text ( i.e. Left , Middle , Right ) .
3. Also in future , we can add the feature of insert of image in the text editor . So it will be more like MS Word.

**References** :

* <https://www.javatpoint.com/java-swing>
* <https://www.javatpoint.com/java-jtextfield>
* <https://www.javatpoint.com/java-jmenuitem-and-jmenu>
* <https://en.wikipedia.org/wiki/Swing_(Java)>
* <https://youtu.be/M03qEx6BYkM>
* <https://www.javatpoint.com/java-jtextpane>
* <https://www.javatpoint.com/java-jpanel>