Unit 1
Abstract Window ToolKit (AWT)

Graphical User Interface

- Graphical User Interface (GUI) offers user interaction via some graphical components.
- For example our underlying Operating System also offers GUI via window, frame, Panel, Button, Textfield, TextArea, Listbox, Combobox, Label, Checkbox etc. These all are known as components. Using these components we can create an interactive user interface for an application.
- GUI provides result to end user in response to raised events. GUI is entirely based events.
- For example clicking over a button, closing a window, opening a window, typing something in a

Advantages of GUI over CUI

- GUI provides graphical icons to interact while the CUI (Character User Interface) offers the simple text-based interfaces.
- GUI makes the application more entertaining and interesting on the other hand CUI does not.
- GUI offers click and execute environment while in CUI every time we have to enter the command for a task.
- New user can easily interact with graphical user interface by the visual indicators but it is difficult in Character user interface.
- GUI offers a lot of controls of file system and the operating system while in CUI you have to use

Contd..

- Windows concept in GUI allow the user to view, manipulate and control the multiple applications at once while in CUI user can control one task at a time.
- GUI provides multitasking environment so as the CUI also does but CUI does not provide same ease as the GUI do.
- Using GUI it is easier to control and navigate the operating system which becomes very slow in command user interface. GUI can be easily customized.

Introduction

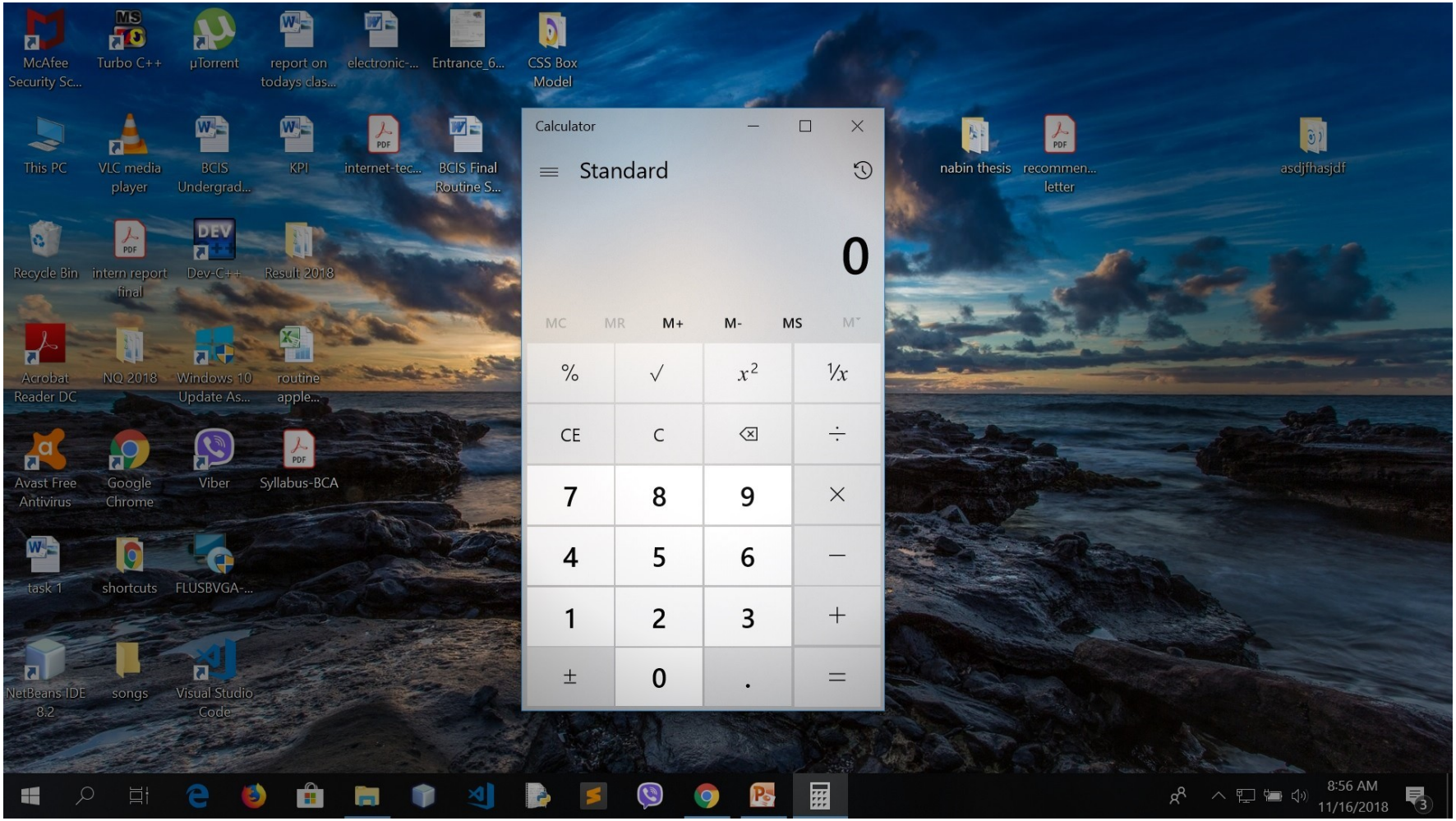
- The Abstract Window Toolkit (AWT) was Java's first GUI framework, and it has been part of Java since version 1.0.
- It contains numerous classes and methods that allow you to create windows and simple controls.
- **Java AWT** (Abstract Windowing Toolkit) is *an API to develop GUI or window-based application in java*.
- Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system.
- AWT is heavyweight i.e. its components uses the resources of system.
- The java.awt package provides classes for AWT api such as TextField, Label, TextArea,

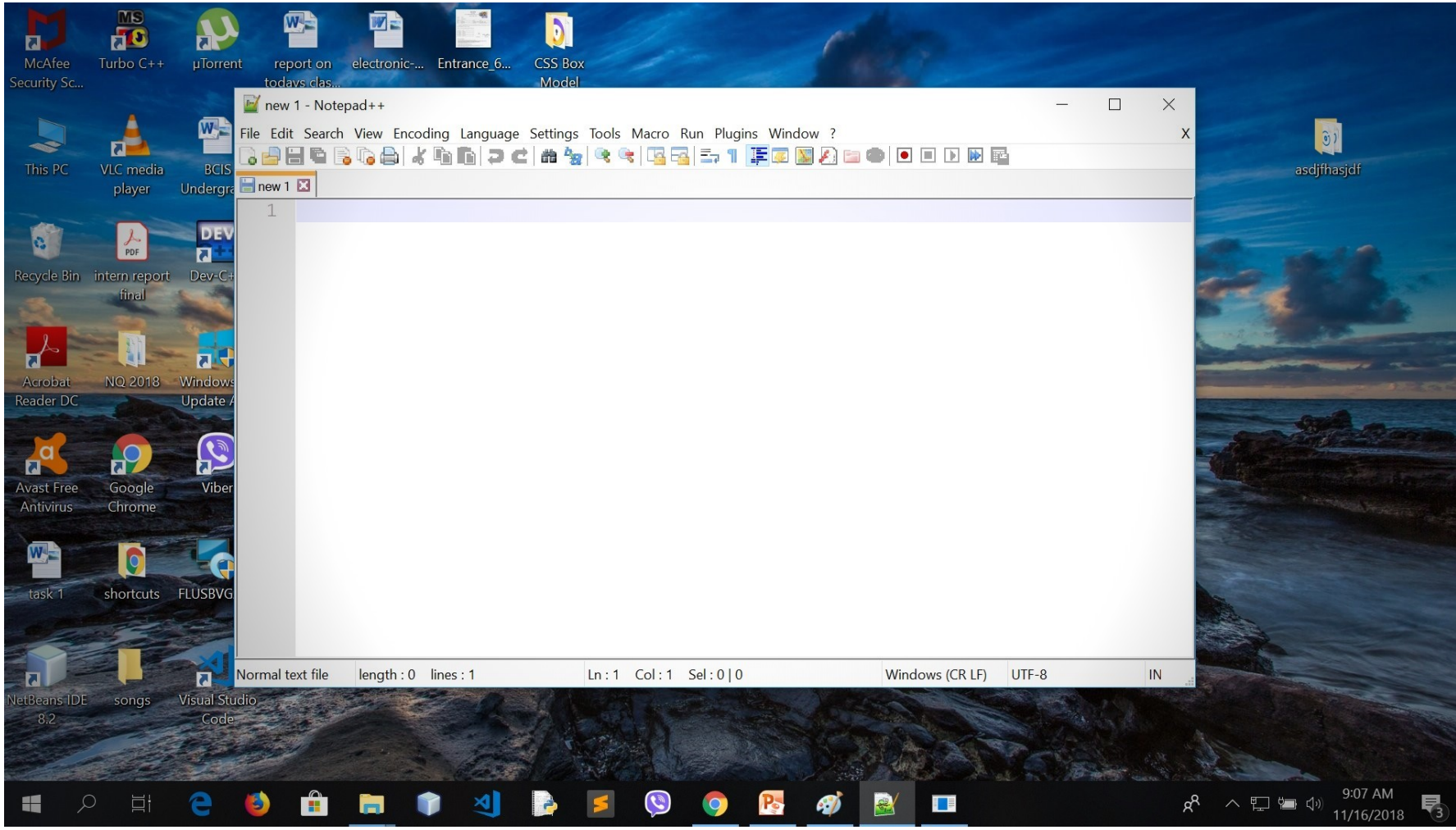
AWT Classes

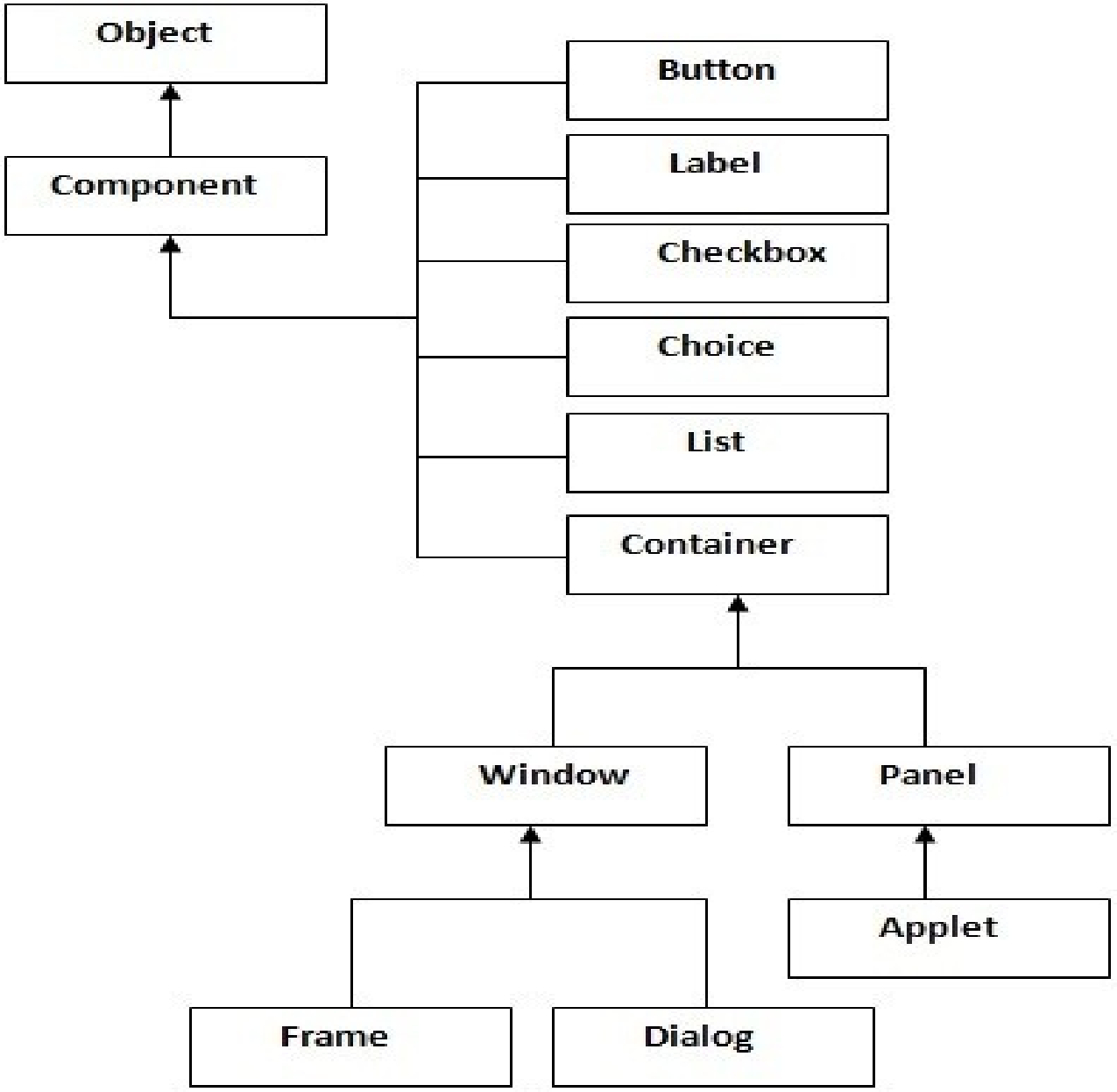
- The AWT classes are contained in the java.awt package.
- It is one of Java’s largest packages.
- Fortunately, because it is logically organized in a top-down, hierarchical fashion.

Window Fundamentals

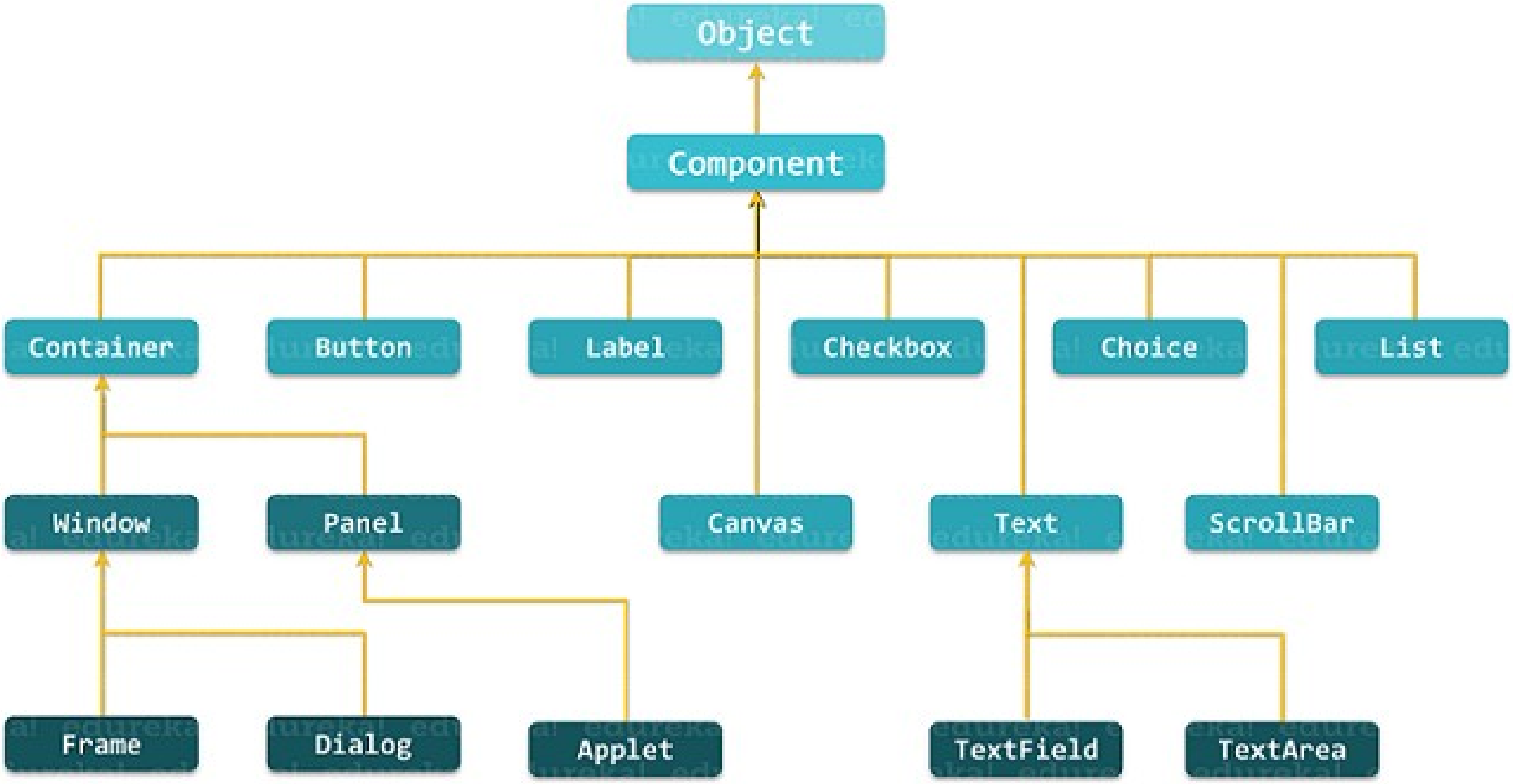
- The AWT defines windows according to a class hierarchy that adds functionality and specificity with each level.
- The two most common windows are those derived from Panel, which is used by applets, and those derived from Frame, which creates a standard application window.
- Much of the functionality of these windows is derived from their parent classes.
- Thus, a description of the class hierarchies relating to these two classes is fundamental to their understanding.


McAfee Security Sc...
Turbo C++
µTorrent
report on todays clas...
electronic-...
Entrance_6...
CSS Box Model
This PC
VLC media player
BCIS Undergrad...
KPI
internet-tec...
BCIS Final Routine S...
nabin thesis
recommen... letter
asdjfhasjdf
Recycle Bin
intern report final
Dev-C++
Result 2018
Acrobat Reader DC
NQ 2018
Windows 10 Update As...
routine apple...
Avast Free Antivirus
Google Chrome
Viber
Syllabus-BCA
task 1
shortcuts
FLUSBVGA-...
NetBeans IDE 8.2
songs
Visual Studio Code
Calculator
Standard
0
MC MR M+ M- MS M˅
% √ x² 1/x
CE C ⌫ ÷
7 8 9 ×
4 5 6 −
1 2 3 +
± 0 . =
8:56 AM
11/16/2018


new 1 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 1
1
Normal text file length : 0 lines : 1 Ln : 1 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 IN
McAfee Security Sc...
Turbo C++
µTorrent
report on todays clas...
electronic-...
Entrance_6...
CSS Box Model
This PC
VLC media player
BCIS Undergra...
asdjfhasjdf
Recycle Bin
intern report final
Dev-C++
Acrobat Reader DC
NQ 2018
Windows Update A...
Avast Free Antivirus
Google Chrome
Viber
task 1
shortcuts
FLUSBVG
NetBeans IDE 8.2
songs
Visual Studio Code
9:07 AM
11/16/2018


Object
Component
Button
Label
Checkbox
Choice
List
Container
Window
Panel
Applet
Frame
Dialog

Hierarchy Of AWT



Component

- At the top of the AWT hierarchy is the Component class.
- Component is an abstract class that encapsulates all of the attributes of a visual component.
- Except for menus, all user interface elements that are displayed on the screen and that interact with the user are subclasses of Component.
- It defines over a hundred public methods that are responsible for managing events, such as mouse and keyboard input, positioning and sizing the window, and repainting.

Container

- The Container class is a subclass of Component.
- It has additional methods that allow other Component objects to be nested within it.
- The Container is a component in AWT that can contain another components like buttons, textfields, labels etc.
- The classes that extends Container class are known as container such as Frame, Dialog and Panel.
- A container is responsible for laying out (that is, positioning) any components that it contains

Window

- The Window class creates a top-level window.
- A top-level window is not contained within any other object; it sits directly on the desktop.
- Generally, you won’t create Window objects directly. Instead, you will use a subclass of Window called Frame, described next.
- The window is the container that have no borders and menu bars. You must use frame, dialog or another window for creating a window.

Panel

- The Panel class is a concrete subclass of Container.
- Panel is the superclass for Applet.
- Panel is a window that does not contain a title bar, menu bar, or border.
- Other components can be added to a Panel object by its
  - add( ) method (inherited from Container).
- Once these components have been added, you can position and resize them manually using the
  - setLocation( ), setSize( ), setPreferredSize( ), or setBounds( ) methods defined by Component.s

Frame

- Frame encapsulates what is commonly thought of as a “window.”
- It is a subclass of Window and has a title bar, menu bar, borders, and resizing corners.
- The Frame is the container that contain title bar and can have menu bars. It can have other components like button, textfield etc.

Useful Methods of Component class

Method	Description
public void add(Component c)	inserts a component on this component.
public void setSize(int width,int height)	sets the size (width and height) of the component.
public void setLayout(LayoutManager m)	defines the layout manager for the component.
public void setVisible(boolean status)	changes the visibility of the component, by default false.

Java AWT Example

- To create simple awt example, you need a frame. There are two ways to create a frame in AWT.
- By extending Frame class (inheritance)
- By creating the object of Frame class (association)

Simple example of AWT by association

```
import java.awt.*;
class First2{
        First2(){
                Frame f=new Frame();

                Button b=new Button("click me");
                b.setBounds(30,50,80,30);
                 // setting button position
                f.add(b);
                f.setSize(300,300);  //
frame size 300 width and 300 height
                f.setLayout(null);  //no layout manager
                f.setVisible(true);
                //now frame will be visible, by default not visible
        }
        public static void main(String args[]){
                First2 f=new First2();
        }
}
```

Simple example of AWT by inheritance

```
import java.awt.*;
class First extends Frame{
        First(){
                Button b=new Button("click me");
                b.setBounds(30,100,80,30);
                // setting button position

                add(b);//adding button into frame
                setSize(300,300);
                //frame size 300 width and 300 height
                setLayout(null);//no layout manager
                setVisible(true);
                //
now frame will be visible, by default not visible
        }
        public static void main(String args[]){
                First f=new First();
}       }
```

Working with Frame Windows

- Here are two of Frame's constructors:
  - Frame( ) throws HeadlessException
  - Frame(String title) throws HeadlessException
- The first form creates a standard window that does not contain a title.
- The second form creates a window with the title specified by title.
- Notice that you cannot specify the dimensions of the window. Instead, you must set the size of the window after it has been created.
- A HeadlessException is thrown if an attempt is made to create a Frame instance in an environment that does not support user interaction.

Setting the Window's Dimensions

- The setSize( ) method is used to set the dimensions of the window. Its signature is shown here:
  - void setSize(int newWidth, int newHeight)
  - void setSize(Dimension newSize)
- The new size of the window is specified by newWidth and newHeight, or by the width and height fields of the Dimension object passed in newSize. The dimensions are specified in terms of pixels.
- The getSize( ) method is used to obtain the current size of a window. One of its forms is shown here:
  - Dimension getSize( )
- This method returns the current size of the window contained within the width and height fields of a Dimension object.

Hiding and Showing a Window

- After a frame window has been created, it will not be visible until you call setVisible( ). Its signature is shown here:
  - void setVisible(boolean visibleFlag)
- The component is visible if the argument to this method is true. Otherwise, it is hidden.

Setting a Window’s Title

- You can change the title in a frame window using setTitle( ), which has this general form:
  - void setTitle(String newTitle)
- Here, newTitle is the new title for the window.

Closing a Frame Window

- When using a frame window, your program must remove that window from the screen when it is closed, by calling setVisible(false).
- To intercept a window-close event, you must implement the windowClosing( ) method of the WindowListener interface.
- Inside windowClosing( ), you must remove the window from the screen.

END OF CHAPTER