

Graphic User Interfaces

COMP163 Introduction to
Computer Programming

“Of all of our inventions for mass communication, pictures still speak the most universally understood language.”

Walt Disney

Quiz

- There will be a quiz in the COMP163 recitation sections from September 11 to September 13
- The quiz will cover all the material since the beginning of the semester
- The questions will be
 - Complete a program
 - Show what a program displays
- There is a sample quiz on Blackboard under course materials

Reading

- Read sections 4.1 through 4.5 of the ZyBooks text and answer all of the participation questions by midnight on Tuesday, September 10, 2019.

Programming Assignment

- A new programming assignment has been posted on Blackboard under Quizzes, Tests and Exams / Programming Assignments
- Upload your Java files to Blackboard before midnight on Friday
- See the tutors in Cherry 124 or your instructor in Cherry 225 for help

String Methods

- There are many methods that can be called on objects of the String class
- `length()` Returns the length of this string

```
String school = "North Carolina A&T State University";  
int howLong = school.length(); // 35
```

Searching Strings

- You can search a string to see if it contains the another string
- `indexOf(String str)` Returns the index within this string of the first occurrence of the specified String or -1 if not found

indexOf Example

- The position of a character in a string starts counting at zero

```
String major = "Computer Science";  
              // 0123456789012345
```

```
int where = major.indexOf( "put" );    // 3
```

```
where = major.indexOf( "e" );         // 6, first match
```

```
where = major.indexOf( "fail" );     // -1
```


Taking Strings Apart

- `substring(int beginIndex, int endIndex)` Returns a new string that is a substring of this string. The substring begins at `beginIndex` and extends to the character at position **`endIndex - 1`**. Thus the length of the substring is `endIndex - beginIndex`
- `charAt(int index)` Returns the `char` value at the specified `index`

substring Example

```
String major = "Computer Engineering";  
              //              1111111111  
              // 01234567890123456789
```

```
String result = major.substring( 11, 14 );
```

result has the value "gin"

Changing Case

- `toUpperCase()` returns a copy of the String with all of the letters in upper case
- `toLowerCase()` returns a copy of the String with all of the letters in lower case

```
String gnu = "Today is 1/27/17";
```

```
String wildebeest = gnu.toUpperCase();
```

```
System.out.println(gnu + "\n" + wildebeest);
```

```
Today is 1/27/17
```

```
TODAY IS 1/27/17
```

Special Characters

- There are some characters that are difficult to type
- Java provides an “escape” characters, the backwards slash, “\”

`\n` new line or the enter key

`\t` tab

`\"` double quote (i.e. "he said, \"Wow\"")

`\\` back slash

- Note that the escape character does not count as a character
- `"\n"` is a String constant with one character

What is the value of result?

```
String major = "Electrical Engineering";  
              // 0123456789012345678901
```

```
String result = major.substring( 4, 8 ) + "k";
```

- A. tric
- B. trick
- C. k
- D. Electrical
- E. none of the above

Miscellaneous String Method

- `trim()` Returns a copy of the string, with leading and trailing whitespace omitted

```
String hair = "  What ever  ";  
                // 01234567890123  
String bald = hair.trim();  
// bald is "What ever";
```

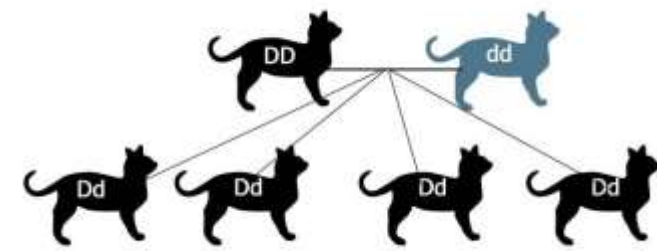
Interesting String Methods

- length
- indexOf
- substring
- charAt
- toUpperCase
- toLowerCase
- trim

Extending Classes

- One of the great advantages of Object Oriented programming is the ability to extend an existing class
- You can create a new class with all of the features of an existing class and then add extra methods or data
- You can extend any class, including the system library classes
- You do not need the source to extend a class

Inheritance



- A new class can inherit all the properties and abilities of another class
- A child class inherits from a parent class
- Any class can be a parent class
- The child class can add extra features or change some features of the parent class
- A class can inherit from a class that inherits from another class to any depth

extends

- In Java you indicate that one class extends or inherits from another by using the **extends** keyword

```
public class NewClass extends OldClass {
```

What is displayed?

```
public class Colors {  
    static void aqua() {  
        System.out.println("Blue");  
    }  
    static void sun(){  
        System.out.println("Yellow");  
    }  
    public static void main(String[] args) {  
        rouge();  
        System.out.println("Gray");  
        aqua();  
    }  
    static void rouge() {  
        System.out.println("Scarlet");  
    }  
}
```

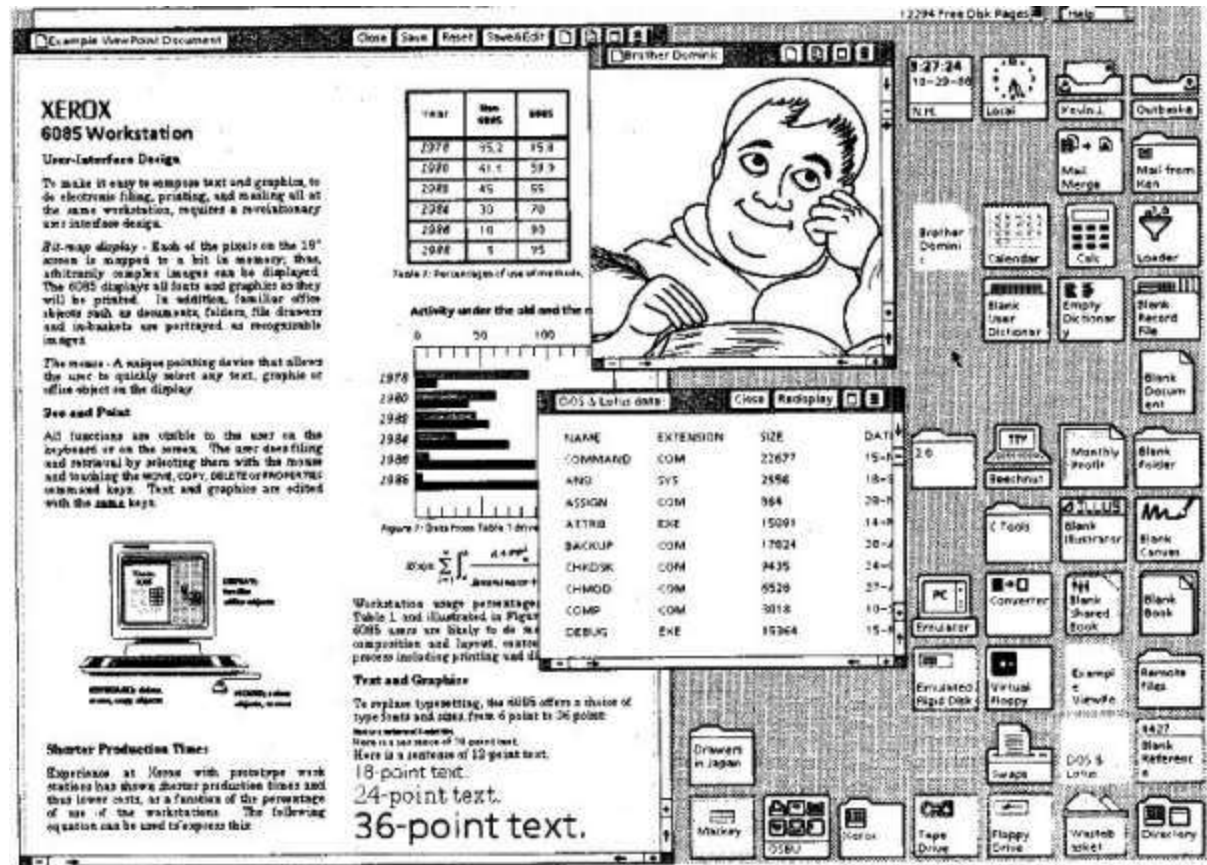
- A. Blue
Yellow
Gray
Scarlet
- B. Scarlet
Gray
Blue
- C. Scarlet
Blue
Gray
- D. Gray
Yellow
Scarlet
- E. none of the above

Graphical User Interface (GUI)

- Almost all of the programs you use have a Graphical User Interface (GUI)
- Before GUIs, all programs ran on the command line with output like `System.out.println` and input like `Scanner`
- Java makes it easy to create GUIs. There are many systems that allow you to create GUIs through a drag and drop interface

History of GUIs

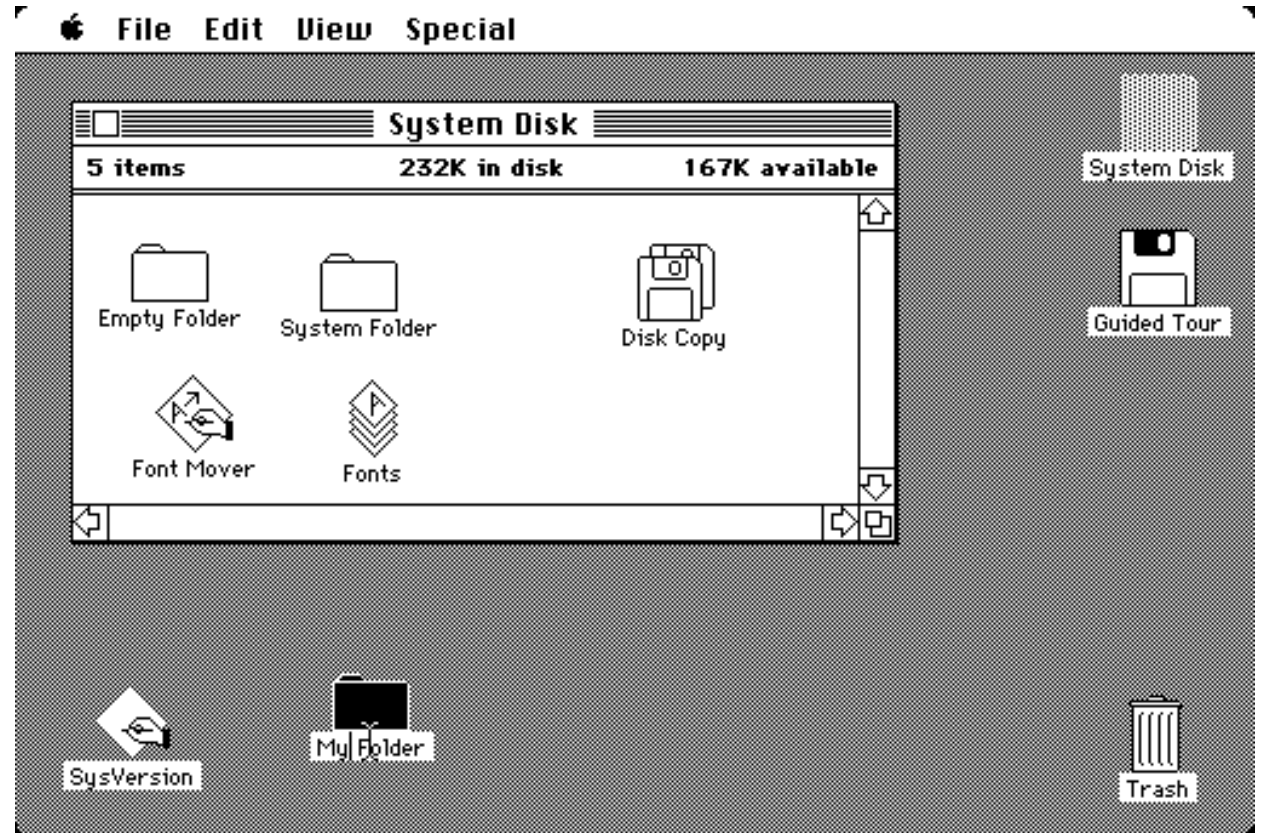
The first general computing system to use a Graphical User Interface was developed at Xerox PARC in 1973. It was never sold as a commercial product.



They developed the WIMP paradigm,
"**W**indow, **I**con, **M**enu, **P**ointing device"

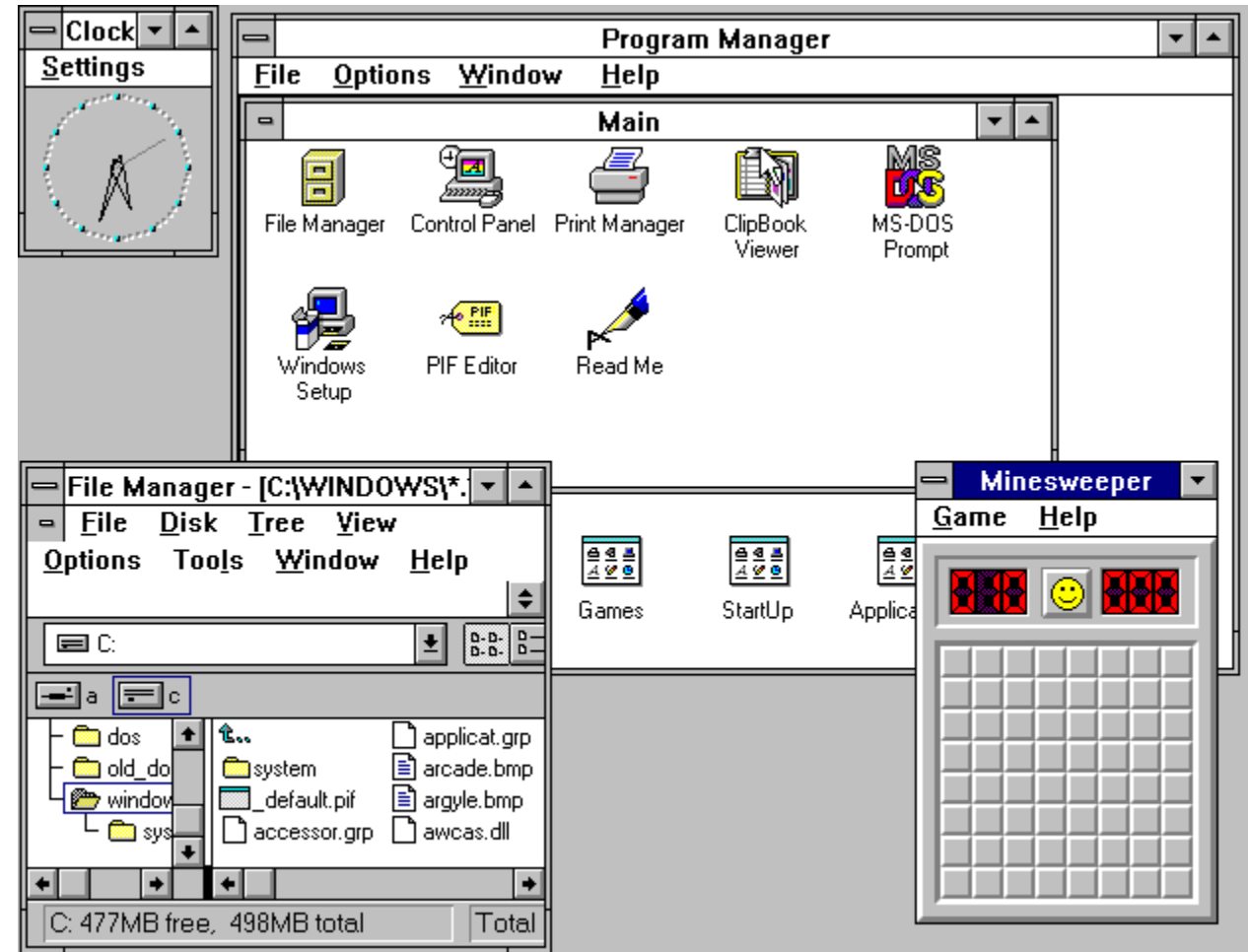
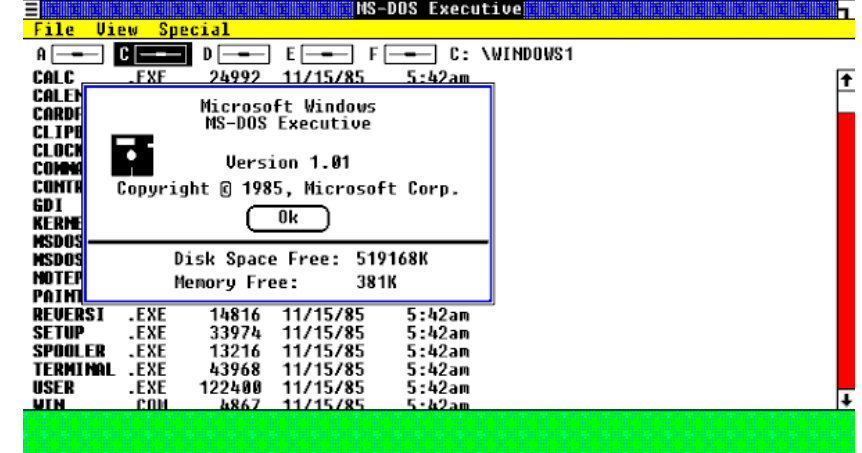
First Commercial GUI

The Apple Macintosh, released in 1984, was the first commercially successful product to use a GUI



Microsoft Windows

Microsoft created Windows 1.0 in 1985. It was ignored until Windows 3.1 in 1990. This was superseded by Windows 95

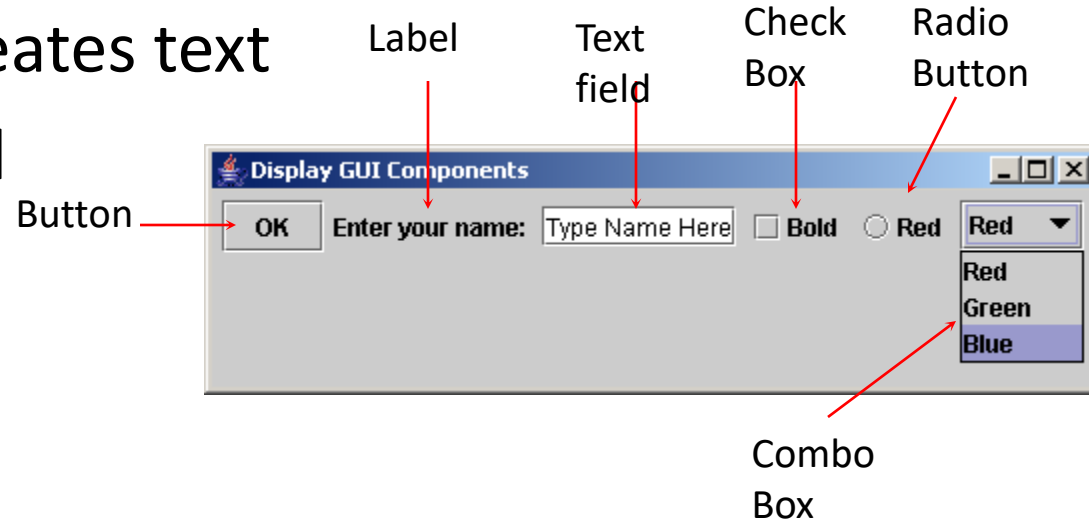


Swing Components

- In the first versions of Java, GUIs were created with the Abstract Windows Toolkit (AWT)
- Later versions of Java introduced the Swing components
- Swing is in `javaX.swing.*`;
- Java also provides the JavaFX system for creating GUIs
- We will be using Swing

Creating GUI Objects

- All of the parts of a GUI are Java objects.
- `javax.swing.JButton` creates buttons
- `javax.swing.JLabel` creates text
- `javax.swing.JTextField` creates input boxes



- `javax.swing.JCheckBox` creates check boxes
- `javax.swing.JRadioButton` creates radio buttons
- `javax.swing.JComboBox` creates menu boxes

Who made the first GUI?

- A. Apple
- B. Sun Microsystems
- C. Microsoft
- D. Xerox

Swing GUI Classes

- There are three categories of Swing classes
 - GUI components
 - Containers
 - GUI helpers

GUI Helper Classes

- Helper classes are used to describe properties of GUI components
- The **Color** class is used to specify the color of objects
- The **Font** class can be used to specify the size and font of text in a GUI

Containers

- Containers are windows that hold the GUI
- **JFrame** is a window that holds GUI objects for a Java application
- **JApplet** is a window in a web browser (such as Internet Explorer or Chrome) that holds a Java applet
- **JDialog** and **JOptionPane** are pop up dialog boxes

Extending JFrame

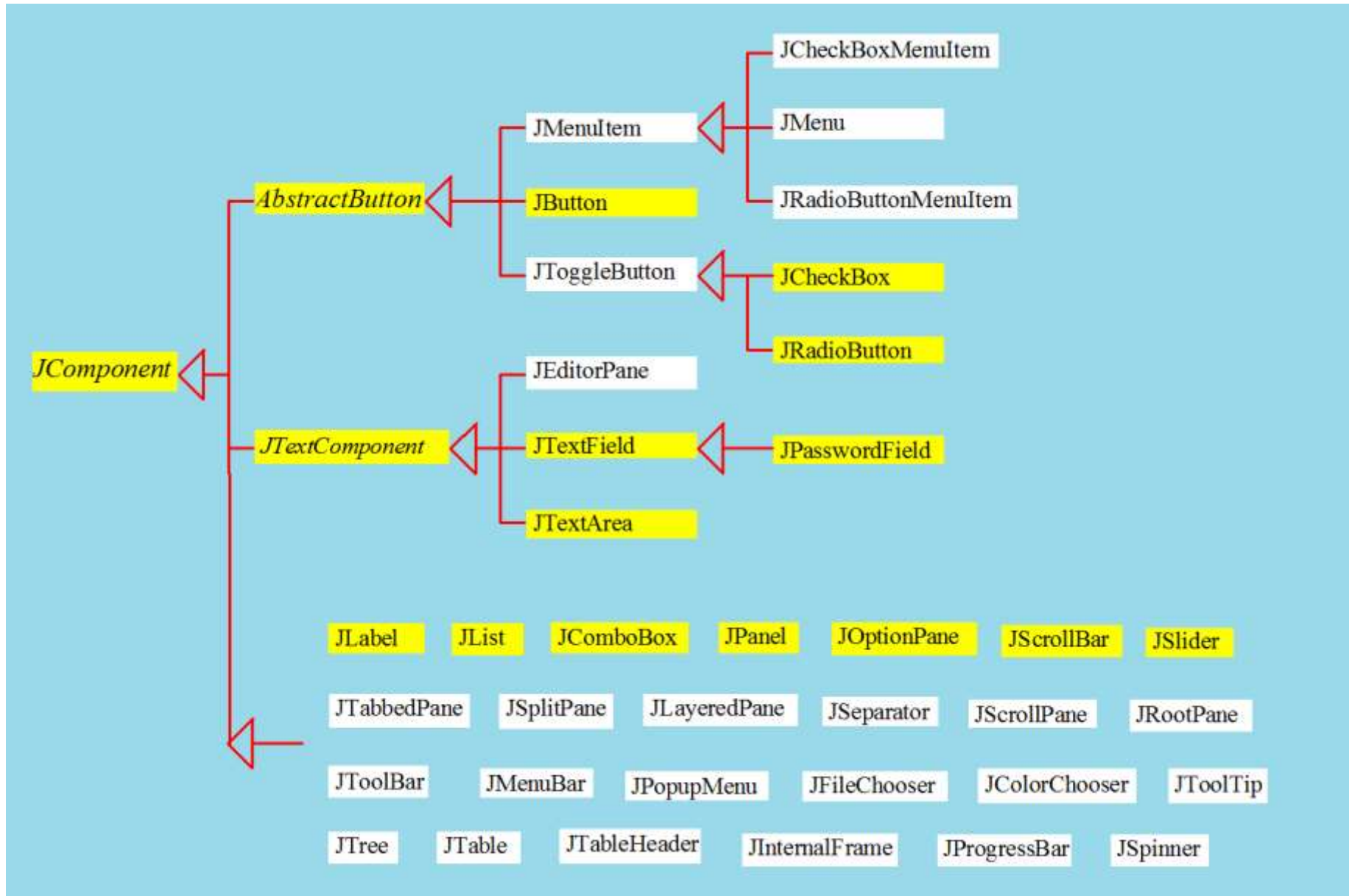
- You can create a GUI by extending javax.swing.JFrame
- The class JFrame provides the “frame” where the GUI will be located

```
public class MyGUI extends javax.swing.JFrame {  
    // program with a GUI  
}
```

Inheritance

- The objects that appear in a GUI (such as buttons, text fields, etc.) all inherit from JComponent
- The JComponent class has lots of methods for specifying the size, shape, location, action and many other general properties of a GUI object
- All GUI objects in the window inherit all of the methods and capabilities of their parent classes, such as JComponent

Class Hierarchy



Consistency

- Because many of the capabilities of a GUI object come from parent classes, you use the same methods to specify these capabilities for all objects
- An object of the class JButton or JTextField or any of the other GUI classes is also a member of the JComponent class

Making an Java GUI

1. Create a class to extend JFrame
2. Create component objects
3. Specify the layout manager
4. Add the component objects to the container
5. Make the GUI visible
6. Specify which objects respond to user input

Constructor Methods

- A constructor method is called when a class is first created
- The constructor method always has the same name as the class (same capitalization)
- The header for a constructor method is


```
public class MyClass {  
    public MyClass () { // constructor  
        // initialization stuff goes here  
    }  
}
```

Creating Objects

- When you create an object, the constructor method is called

Class

Constructor method


`MyClass turtle = new MyClass();`

- If you do not write a constructor method, Java will automatically create one that does nothing

JFrame Program

- A GUI extends the class JFrame
- We will be using the Java Swing classes which contains the class JFrame that inherits from java.awt.Frame
- A GUI should have a constructor method

```
public MyProg() {
```

- This is called when the program is first executed

Creating a JFrame

```
public class FirstGUI extends javax.swing.JFrame {  
  
    public FirstGUI () {  
        setSize(250, 200);    // set the frame size  
    }  
}
```

Setting the Frame Size

- You can specify how big the frame will be with the **setSize(xSize, ySize)** method
- The sizes are specified in pixels the same as the graphics methods used
- When the program is executed, this specifies how much space it will occupy on the screen

Content Pane

- A JFrame has a content pane that holds the visible components of the frame, like buttons
- Programs need to get the content pane using the `getContentPane()` method of JFrame

```
java.awt.Container  goat = getContentPane();
```


Creating a GUI

```
public class FirstGUI extends javax.swing.JFrame {  
  
    public FirstGUI () {  
        setSize(250, 200);  
        java.awt.Container goat = getContentPane();  
    }  
}
```

GUI components are Java

- A. methods
- B. classes
- C. objects
- D. JFrames
- E. types

Creating Components

- Most Swing components have a constructor that takes one String parameter specifying the text to be displayed in that component

```
javax.swing.JButton dog =  
    new javax.swing.JButton( "OK" );  
javax.swing.JLabel cat =  
    new javax.swing.JLabel( "Hi there" );
```

Most Useful Components

- **JButton** – Useful to tell the program to do something
- **TextField** – Read text entered by the user
- **JLabel** – Display text or solutions on the screen

Components as Instance Variables

- The GUI component variables should be defined in the class and **not** in the constructor method
- Calls to methods of the components are often put in the constructor method of a frame

Creating Components

```
public class FirstGUI extends javax.swing.JFrame {  
    javax.swing.JButton dog =  
        new javax.swing.JButton( "Go" );  
  
    javax.swing.JLabel cat =  
        new javax.swing.JLabel("COMP163");  
  
    public FirstGUI () {  
        setSize(250, 200);  
  
        java.awt.Container goat = getContentPane();  
    }  
}
```

Putting Them Where You Want Them

- The **Layout Manager** determines where components will appear in a GUI
- There are several ways to specify position and size
- Some are more portable than others
- Some are more frustrating than others

Layout Managers

- BorderLayout
- FlowLayout
- GridLayout
- BorderLayout
- Absolute Positioning

BoxLayout

- The BoxLayout is relatively simple
- It positions components as “boxes” either top to bottom (Y_AXIS) or left to right (X_AXIS)
- You must create a BoxLayout object

```
javax.swing.BoxLayout whale = new  
    javax.swing.BoxLayout(goat, javax.swing.BoxLayout.Y_AXIS);
```

or

```
import javax.swing.*;
```

```
BoxLayout whale = new BoxLayout(goat, BoxLayout.Y_AXIS);
```

Specifying the Layout Manager

- You must specify what layout manager your program will use with the **setLayout** method
- The parameter of the **setLayout** method is any layout manager object

```
javax.swing.BoxLayout whale = new  
    javax.swing.BoxLayout(goat, javax.swing.BoxLayout.Y_AXIS);
```

```
setLayout( whale );
```

Adding Layout Manager

```
public class FirstGUI extends javax.swing.JFrame {
    javax.swing.JButton dog =new javax.swing.JButton( "Go" );
    javax.swing.JLabel cat =new javax.swing.JLabel("COMP163");
    public FirstGUI () {
        setSize(250, 200);
        java.awt.Container goat = getContentPane();
        javax.swing.BoxLayout whale = new
            javax.swing.BoxLayout(goat,javax.swing.BoxLayout.Y_AXIS);
        setLayout( whale );
    }
}
```

Adding Components to a GUI

- To make them appear in a frame the components must be added to the JFrame
- The add method links a component to a content pane object

```
goat.add( GUIthing );
```

- The order in which the components are added determines the order in which they will appear in the GUI

Adding Components to Pane

```
public class FirstGUI extends javax.swing.JFrame {  
    javax.swing.JButton dog = new javax.swing.JButton( "Go" );  
    javax.swing.JLabel cat = new javax.swing.JLabel("COMP163");  
    public FirstGUI () {  
        setSize(250, 200);  
        java.awt.Container goat = getContentPane();  
        javax.swing.BoxLayout whale = new  
        javax.swing.BoxLayout(goat,javax.swing.BoxLayout.Y_AXIS);  
        setLayout( whale );  
        goat.add( cat );  
        goat.add( dog );  
    }  
}
```

setDefaultCloseOperation

- Everyone knows that a program will terminate if you press the red **X** in the upper right corner
- In Java pressing the red X calls a method
- A program can do whatever you want it to do when you press the red X

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE) ;
```

- This makes the Java program terminate when you press the red X

setVisible

- Most GUI components have a setVisible method

`setVisible(true or false);`

- If the boolean parameter is true, the object shows, if false it does not show
- Calling **setVisible (true)** makes the frame appear onscreen

Making it Visible

```
public class FirstGUI extends javax.swing.JFrame {  
    javax.swing.JButton dog = new javax.swing.JButton( "Go" );  
    javax.swing.JLabel cat = new javax.swing.JLabel("COMP163");  
    public FirstGUI () {  
        setSize(250, 200);  
        java.awt.Container goat = getContentPane();  
        javax.swing.BoxLayout whale = new  
            javax.swing.BoxLayout(goat, javax.swing.BoxLayout.Y_AXIS);  
        setLayout( whale );  
        goat.add( cat );  
        goat.add( dog );  
        setDefaultCloseOperation(javax.swing. JFrame.EXIT_ON_CLOSE);  
        setVisible( true );  
    }  
}
```


Main Method

- The main method of a GUI is very simple
- It only creates an object of your program's class

```
public class MyProg {  
    public static void main(String[] goat) {  
        MyProg cow = new MyProg();  
    }  
}
```

A Simple GUI Program

```
public class FirstGUI extends javax.swing.JFrame {
    javax.swing.JButton dog = new javax.swing.JButton( "Go" );
    javax.swing.JLabel cat = new javax.swing.JLabel("COMP163");
    public FirstGUI () {
        setSize(250, 200);
        java.awt.Container goat = getContentPane();
        javax.swing.BoxLayout whale = new
            javax.swing.BoxLayout(goat,javax.swing.BoxLayout.Y_AXIS);
        setLayout( whale );
        goat.add( cat );
        goat.add( dog );
        setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
        setVisible( true );
    }
    public static void main(String[] rat) {
        FirstGUI myprog = new FirstGUI();
    }
}
```

Quiz

- There will be a quiz in the COMP163 recitation sections from September 11 to September 13
- The quiz will cover all the material since the beginning of the semester
- The questions will be
 - Complete a program
 - Show what a program displays
- There is a sample quiz on Blackboard under course materials

Reading

- Read sections 4.1 through 4.5 of the ZyBooks text and answer all of the participation questions by midnight on Tuesday, September 10, 2019

Programming Assignment

- A new programming assignment has been posted on Blackboard under Quizzes, Tests and Exams / Programming Assignments
- Upload your Java files to Blackboard before midnight on Friday
- See the tutors in Cherry 124 or your instructor in Cherry 225 for help