

AP Computer Science
Crib Sheet for GUI material in Unit 1 (Chapters 1 – 4)

Class	JFrame
Purpose	<i>This is a popup window frame into which we can put content</i>
Import	javax.swing.JFrame
Constructor	JFrame theGUI = new JFrame();
Methods	void setTitle(String s) void setSize(int x, int y) void setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE) void setVisible(boolean b) void setResizable(boolean b) Container getContentPane() void pack()
Class	JPanel
Purpose	<i>This is a generic "canvas" on which we draw – we place these in JFrames</i>
Import	javax.swing.JPanel
Constructor	JPanel panel = new JPanel();
Methods	void setBackground(Color c) void paintComponent(Graphics g) // we overwrite this, see example void setPreferredSize(new Dimension(int width, int height)) int getWidth() int getHeight()
Class	Color
Purpose	<i>This keeps track of color information in 8-bit red/green/blue format</i>
Import	java.awt.Color
Constructor	Color c = new Color(255, 177, 0); // note red, green, blue
Class	Container
Purpose	<i>This is a useful generic device we use whenever we need to put a bunch of things inside something else – we need these when doing a layout of many JPanels</i>
Import	java.awt.Container
Constructor	Container pane = JFrame.getContentPane();
Methods	void add(JPanel panel) void add(JPanel panel, BorderLayout.NORTH) // for example void setLayout(new GridLayout(int rows, int columns) // for example
Class	Font
Purpose	<i>This stores information about font type when drawing Strings</i>
Import	java.awt.Font
Constructor	Font f = new Font("Courier", Font.BOLD, 14);
Class	JOptionPane
Purpose	<i>This creates a popup window in which the user can enter in data</i>
Import	javax.swing.JOptionPane
Implement.	String input = JOptionPane.showInputDialog("Your age", 16); int age = Integer.parseInt(input); // or could use double a = Double.parseDouble(input); JOptionPane.showMessageDialog(null, "Good for you!");

AP Computer Science
Crib Sheet for GUI material in Unit 1 (Chapters 1 – 4)

Class	Graphics
Purpose	A “graphics context” is a tool for drawing & writing in a JPanel
Import	java.awt.Graphics
Constructor	no need - JPanel sends a Graphics object to paintComponent
Methods	void setColor(Color c) void setFont(Font f) void drawString(String s, int x, int y) void drawLine(int x1, int y1, int x2, int y2) void drawRect(int x, int y, int width, int height) void drawOval(int x, int y, int width, int height) void fillRect(int x, int y, int width, int height) void fillOval(int x, int y, int width, int height)
Note	The coordinate system starts at the upper-left corner, with +x towards the right and +y downward

// EXAMPLES OF THE BASIC PAIR OF PROGRAMS FOR DOING YOUR OWN GRAPHICS

// sample GUIWindow program with main() method - you execute this

```
import javax.swing.*;
import java.awt.*;
public class GUIWindow {
    public static void main(String [] args) {
        JFrame theGUI = new JFrame();
        theGUI.setTitle("GUI Program");
        theGUI.setSize(300, 200);
        theGUI.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        ColorPanel panel = new ColorPanel(Color.white);
        Container pane = theGUI.getContentPane();
        pane.add(panel);
        theGUI.setVisible(true);
    }
}
```

// sample ColorPanel program with no main() method - you alter this

```
import javax.swing.*;
import java.awt.*;
public class ColorPanel extends JPanel {
    public ColorPanel(Color backColor) {
        setBackground(backColor);
    }
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.blue);
        g.drawRect(10, 5, 120, 20);
        g.setColor(Color.red);
        g.drawString("Hello world!", 20, 20);
    }
}
```