

AP Computer Science  
Crib Sheet for GUI material

**Class JFrame**

*Purpose* This is a popup window frame into which we can put content

*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* This is a generic "canvas" on which we draw – we place these in JFrames

*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() // gives you width of panel  
int getHeight()  
void repaint() // calls the paintComponent() method

**Class Color**

*Purpose* This keeps track of color information in 8-bit red/green/blue format

*Import* java.awt.Color

*Constructor* Color c = new Color(255, 177, 0); // note red, green, blue

**Class Container**

*Purpose* 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.

*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* This stores information about font type when drawing Strings

*Import* java.awt.Font

*Constructor* Font f = new Font("Courier", Font.BOLD, 14);

**Class JOptionPane**

*Purpose* This creates a popup window in which the user can enter in data

*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

<b>Class</b>	<b>Graphics</b>
<b>Purpose</b>	A "graphics context" is a tool for drawing & writing in a JPanel
<b>Import</b>	java.awt.Graphics
<b>Constructor</b>	no need - JPanel sends a Graphics object to paintComponent
<b>Methods</b>	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)
<b>Note</b>	The coordinate system starts at the upper-left corner, with +x towards the right and +y downward

// 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);
    }
}
```

AP Computer Science  
Crib Sheet for GUI material

<b>Class</b>	<b>ImageIcon</b>
<b>Purpose</b>	<i>This is a place in memory to put an uploaded image</i>
<b>Import</b>	javax.swing.ImageIcon
<b>Constructor</b>	ImageIcon image = new ImageIcon(String filename)
<b>Methods</b>	void paintIcon(JPanel p, Graphics g, int x, int y)
<b>Note</b>	Within the overridden paintComponent method of an extended JPanel class (such as ColorPanel), you would use the command <code>image.paintIcon(this, g, 50, 50)</code> ; in this case, 'this' is a keyword referring to the ColorPanel object you are in, g is the graphics context, and the two coordinates represent the position of the upper-left-hand corner

// 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);
        ImageIcon image = new ImageIcon("filename.jpg");
        ColorPanel panel = new ColorPanel(Color.white, image);
        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 {
    private ImageIcon image;
    public ColorPanel(Color backColor, ImageIcon i) {
        setBackground(backColor);
        image = i;
    }
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        image.paintIcon(this, g, 50, 50);
    }
}
```

AP Computer Science  
Crib Sheet for GUI material

**Class**            **MouseAdapter**

**Purpose**            *This is a base class we extend to make "listeners" that respond to mouse movements*

**Import**            java.awt.event.MouseAdapter

**Constructor**      see sample program the follows

**Methods**          void mouseEntered(MouseEvent e)  
void mouseExited(MouseEvent e)  
void mousePressed(MouseEvent e)  
void mouseReleased(MouseEvent e)  
void mouseClicked(MouseEvent e)

**Class**            **MouseMotionAdapter**

**Purpose**            *This is a base class we extend to make "listeners" that respond to mouse movements*

**Import**            java.awt.event.MouseMotionAdapter

**Constructor**      see sample program that follows

**Methods**          void mouseMoved(MouseEvent e)  
void mouseDragged(MouseEvent e)

**Class**            **MouseEvent**

**Purpose**            *These objects contain information about things that happen with the mouse.*

**Import**            java.awt.event.MouseEvent

**Constructor**      see sample program that follows

**Methods**          int getX()  
int getY()  
int getButton()

// sample class that displays the current mouse position

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class MousePanel extends JPanel {
    int x, y;
    public MousePanel() {
        x = 100;
        y = 100;
        addMouseListener(new PanelListener());
    }
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawString("(" + x + ", " + y + ")", x, y);
    }
    private class PanelListener extends MouseAdapter {
        public void mousePressed(MouseEvent e) {
            x = e.getX();
            y = e.getY();
            repaint();
        }
    }
}
```