**Lesson 2 – Swing: Layout Managers**

* To make our GUIs visually appealing our components must be arranged in a logical order
  + Done using **Layout Manager**
  + **Types of Layouts given to us by Swing**
    - **Flow Layout (Default)**
    - **Grid Layout**
    - **Border Layout**
    - **Box Layout**
    - **GridBag Layout**
* To use a Layout Manager, we must do 2 things
  + **Instantiate** a LayoutManager Object
    - LayoutIChose variableName = new LayoutIChose();
  + Set the layout to the content pane
    - myPanel.setLayout(variableName);
  + You can now add your components to the JPanel
* **Flow Layout**
  + This is the default layout strategy
  + This layout adds components in a line until the end of the row (left -> right)
    - this.add(this.component);
  + Length of a row depends on width of JPanel and preferred size of components
  + Individual rows are centered
    - Can be changed if desired
* **Grid Layout**
  + All components are laid out in a grid of x rows and y columns
  + X and Y are determined on creation of the layout object
  + Create a Grid Layout Object by:
    - *GridLayout layout = new GridLayout(int x, int y);*
  + Components are added left to right, top to bottom
  + All components are the same size (ignoring preferred size)
* **Border Layout (Usually most useful)**
  + Adds up to 5 components
  + Divides into 5 different areas
    - North, West, South, East, and Center
    - North and South = Entire width
      * Height is determined by the preferred height
    - West and East cover the remaining height
      * The width is determined by the preferred size
    - Center expands to fill any remaining space
  + Positions without any components do not take any space
    - E.g. if EAST does not take up any space then center will fill it
  + Components are added by specifying their desired location
    - *panel.add(component, BorderLayout.NORTH);*
* **Box Layout**
  + Allows us to arrange components in either
    - Horizontal Row
    - Vertical Row
  + *BoxLayout layout = new BoxLayout(Container target, int axis);*
    - *Where axis is either X\_AXIS or Y\_AXIS*
  + Component size in the opposite orientation matches the largest preferred size
  + Components are added left to right or top to bottom (depending on axis)
* **GridBag Layout**
  + The most flexible and most complex Layout Manager
  + Places components in a grid from left to right and top to bottom
  + Each cell in the grid can vary in size
  + Each component can take up more than 1 cell
  + Requires the use of GridBagConstraints objects
* **Creating Complex Layouts**
  + Complexity can be added by **adding panels into panels**
    - EXAMPLE:
      * Place a GridLayout in the East position of a BorderLayout