|  |  |
| --- | --- |
| **Computer Science 1** | **Lab 07B**  **Multi-Day Major Python Assignment** |
| **Selection with Graphics** | **70, 80, 90, 100 & 110 Point Versions** |
| **Assignment Purpose:**  The purpose of this program is to demonstrate knowledge of Multi-Way Selection and how it can be integrated with Graphics and GUI Keyboard Input. | |

Write a program that will use graphics-based input that allows the user to choose different graphics shapes and colors and then displays the selected image. The program essentially starts with this provided **numinput** command:

**14 shapeNum = numinput("Shape Selection", "1 = Line \n2 = Rectangle \n3 = Circle \n4 = Oval \n5 = Regular Octagon \n6 = Star \n7 = Burst")**

|  |  |
| --- | --- |
| which creates this input window: | NOTE: It is the **\n** “new line” *Escape Sequences* that allow the GUI input window to have a multi-line prompt.  It is not required that your program displays the exact same shapes that are listed in the GUI input window on the left. Feel free to edit line #14 if you wish to display different shapes. Keep in mind that while you are allowed to change the shapes, you still are required to display 7 completely different shapes.  Also, your shapes can involve just about any of the draw/fill commands in the **Graphics** library – even ones that we have not learned yet in class – however, do not choose **drawPixel** or **drawPoint**. A dot is simply not enough.  HINT: If you are having difficulty getting started, you should refer to program example **Selection13.py**. |

|  |  |
| --- | --- |
| **Lab 07B Student Version** | **Do not copy this file, which is provided.** |
| **1 # Lab07Bst.py  2 # "Selection With Graphics"  3 # This is the student, starting version of Lab 07B.  4   5   6   7 from Graphics import \*  8   9 beginGrfx(1300,700) 10  11 # Substitute your own name here. 12 drawHeading("John Smith","7B") 13  14 shapeNum = numinput("Shape Selection", "1 = Line \n2 = Rectangle \n3 = Circle \n4 = Oval \n5 = Regular Octagon \n6 = Star \n7 = Burst") 15  16  17  18  19 endGrfx()  20** | |

**70 Point Version Specifics and Sample Output**

The 70-point version is strictly concerned with the ability to display any of the shapes listed in the “Shape Selection” menu. Remember, you need at least 7 different shapes, but they do not need to be the exact same shapes as mine. On the next page, you will see one issue with this version.

|  |  |
| --- | --- |
|  |  |

**70 Point Version Specifics and Sample Output Continued**

Note, if the user enters a value that is outside of the proper range, nothing is displayed.

This is fine for the 70-point version.

|  |  |
| --- | --- |
|  |  |

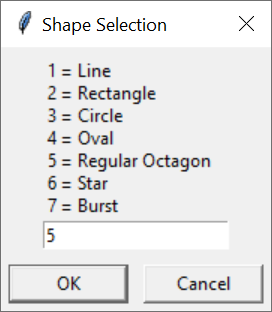
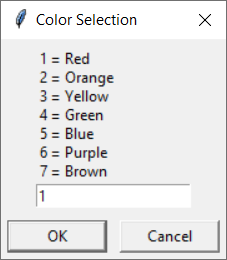
**80 Point Version Specifics and Sample Output**

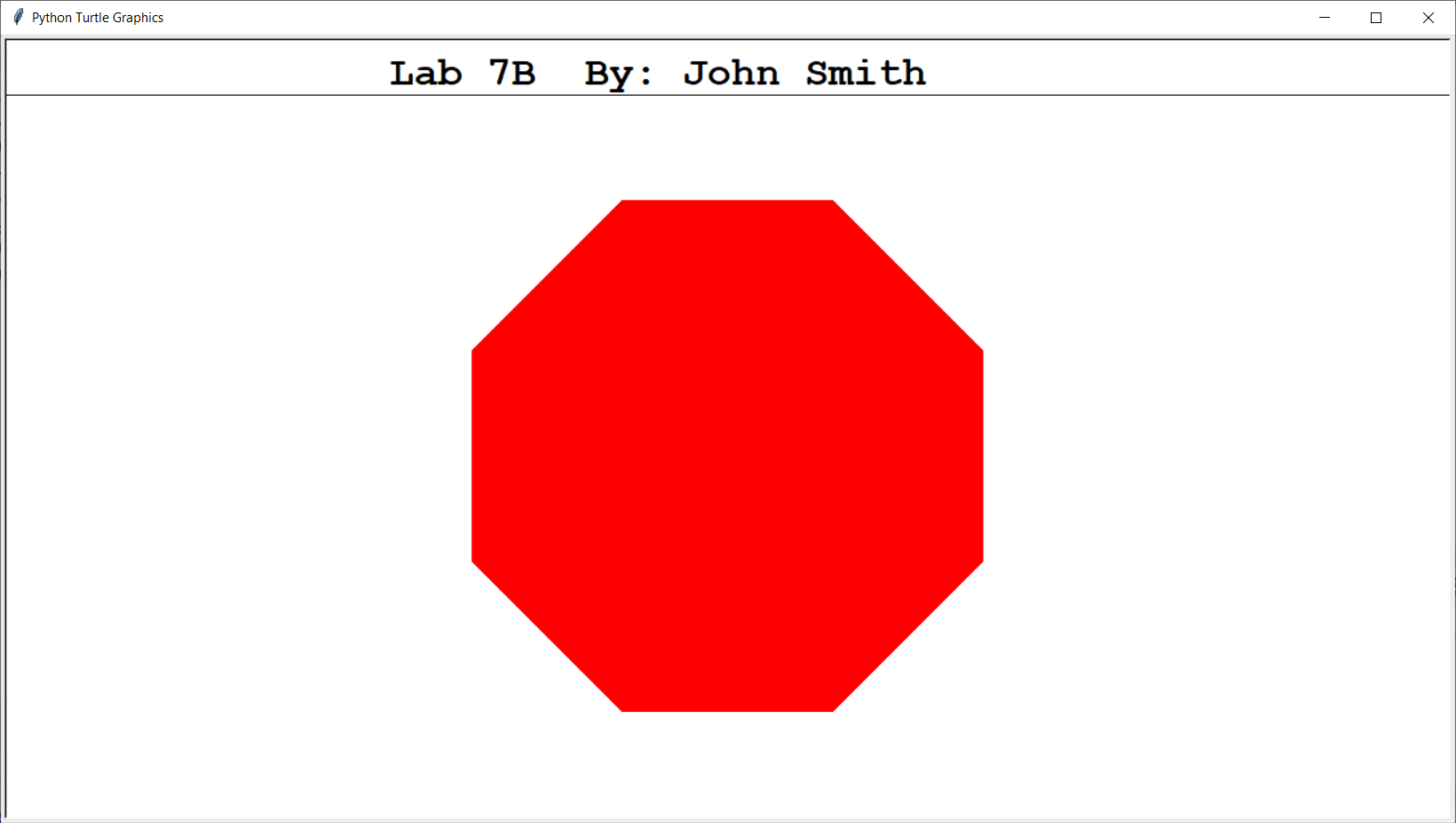
The 80-point version displays the same shapes as the 70-point version; however, if the user enters a value that is outside of the proper range, an error message, like the one below, needs to be displayed.

|  |  |
| --- | --- |
|  |  |

**90 Point Version Specifics and Sample Output**

The 90-point version requires everything from the 80-point version and adds a second input window to select among a minimum of 7 different colors. You do not need to use the exact same colors that I have in my example below. You can use any 7 colors.

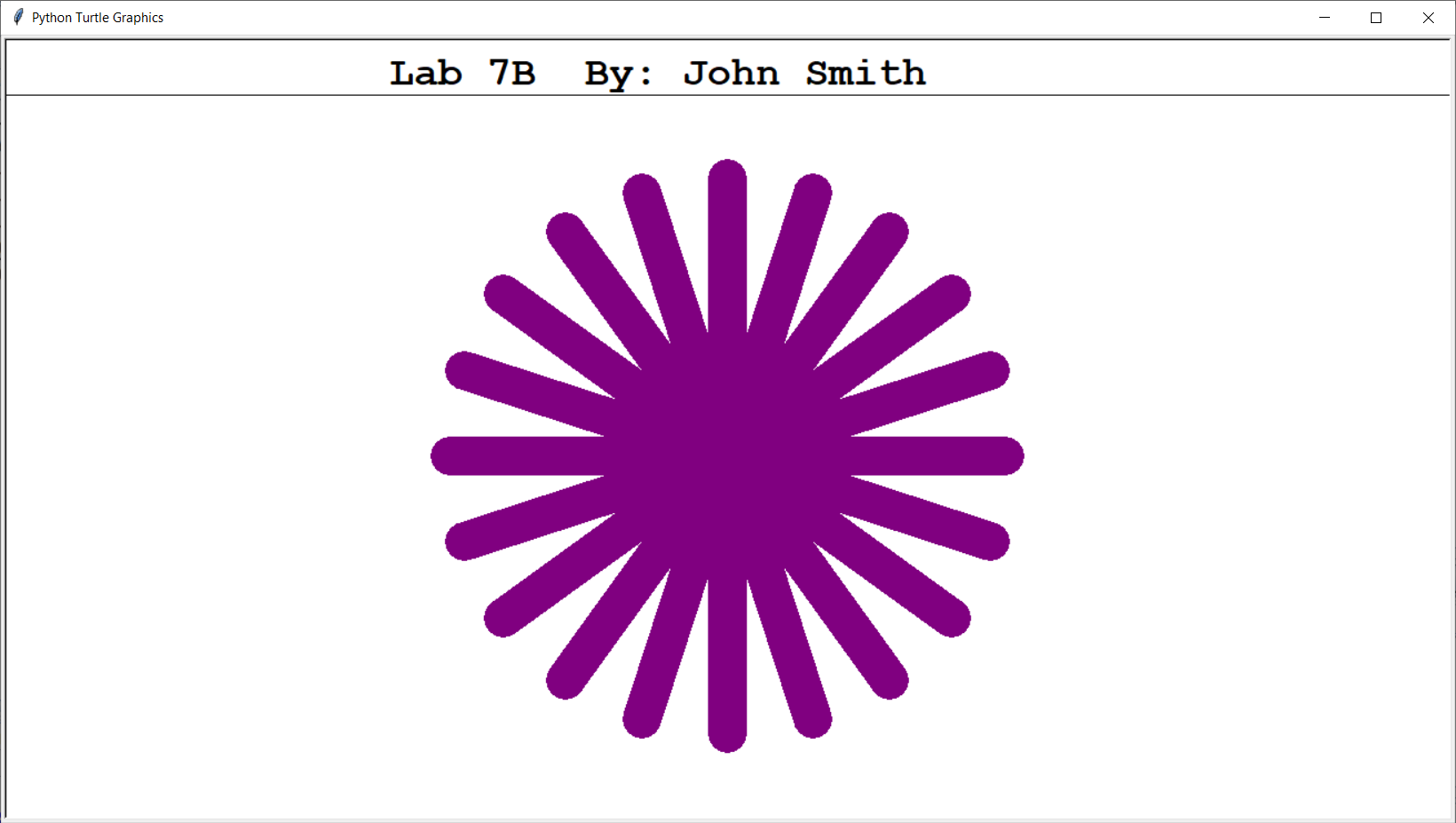


NOTE: It does not matter which menu is displayed first; however, you need to make sure you set the color BEFORE you draw the shape. Setting the color afterward is too late.

**100 Point Version Specifics and Sample Output**

The 100-point version requires everything from the 90-point version and adds a third input window to enter the thickness of the lines. To properly test this effect, you should select a shape that is merely “drawn” and not “filled”.

|  |  |  |
| --- | --- | --- |
|  |  |  |

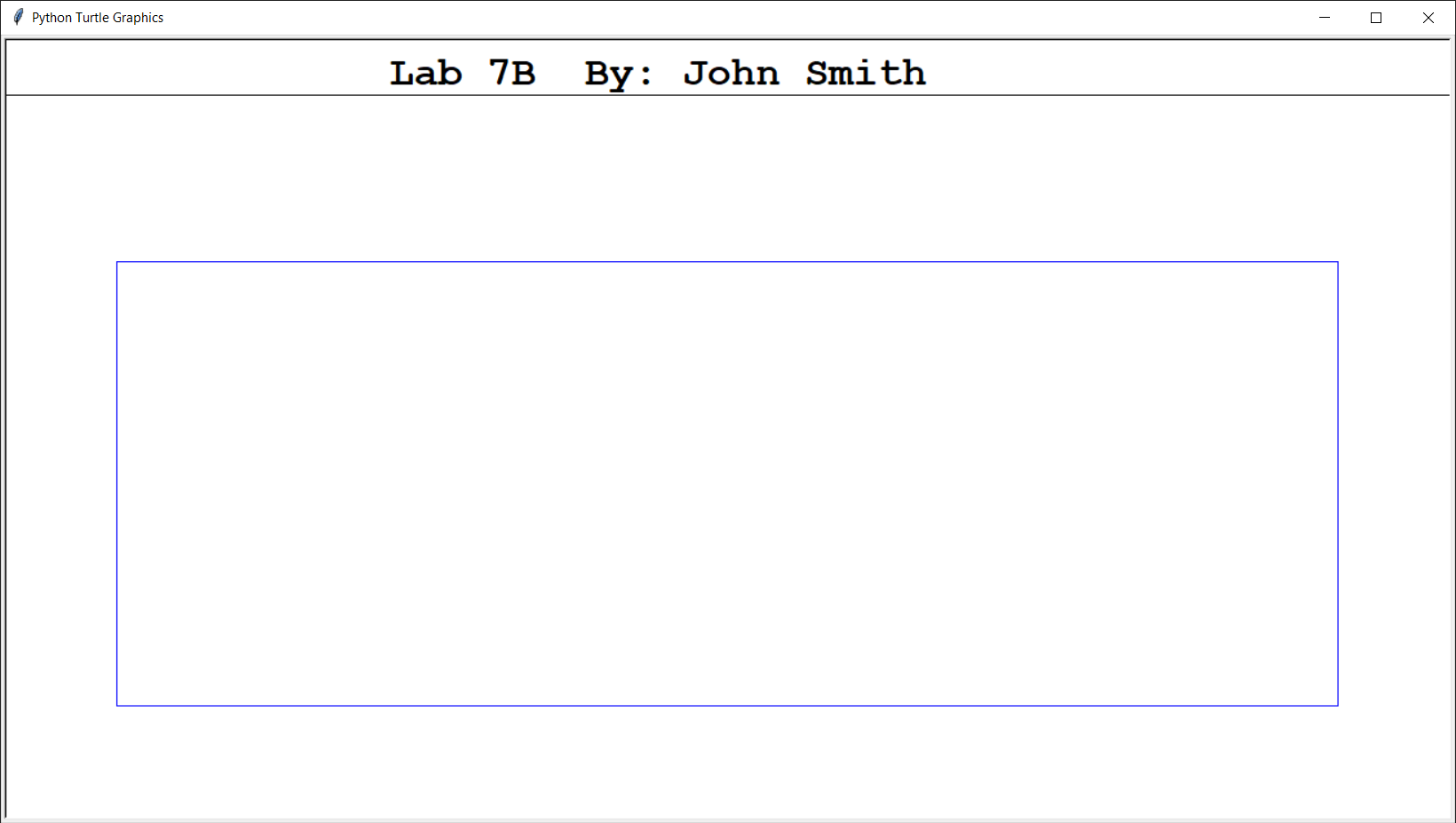


NOTE: As with the 90-point version, it does not matter which menu is displayed first; however, you need to make sure you set BOTH the width AND the color BEFORE you draw the shape. Setting the width and/or the color afterward is too late.

**110 Point Version Specifics and Sample Output #1**

The 110-point version requires everything from the 100-point version and adds a fourth input window to allow the user the ability to decide if they want their enclosed shapes – like rectangles, circles, ovals, polygons and stars – to be filled or not. The user will enter a CAPITAL ‘**Y**’ or ‘**N**’ for this input. For example, suppose the selected shape is a “Rectangle”. If the user entered ‘**Y**’ on this menu, the **fillRectangle** command would be executed; otherwise the **drawRectangle** command is executed.

|  |  |  |
| --- | --- | --- |
|  |  |  |



**110 Point Version Sample Output #2**

|  |  |  |
| --- | --- | --- |
|  |  |  |

