COSC 1P02 Tutorial Exercise 1
Sept 23rd and 27th, 2022

Nesting & Object References

In this exercise we will study the difference between nested and non-nested loop structures and the significance of objects being accessed by reference variables. By the end of this activity you should have a clear understanding of the behavioral differences between sequential code—loops that are written one after the other in sequence—and nested code—loops that are written one within the other. It should be possible for you to determine the exact number of times code is executed in each of these cases. You will also understand the consequences of an object (e.g. a turtle) being referenced by a reference variable (i.e. a variable containing an address) and the effect of reference assignment.

Use `E1 Answer.pdf` *for your own notes.*

Part A

To demonstrate loop behavior we will use a simple graphing routine. For every cycle of a loop, a pulse will be drawn on the graph.

Open the project attached in BlueJ by clicking on the `package` file in the folder. When you run the program, you should be presented with two classes: `Pulses` and `TwoByTwo`. To run a class, right-click and select either the first option or second option (both will do that same thing). First, run `Pulses`. With a `TurtleDisplayer` on which a number of pulses, some large and some small, are drawn. What happened (i.e. how many pulses and of what size?). *Enter your answers on the answer sheet.*

Consider the source code of the `Pulses` program. In the constructor you will notice 2 for loops. The first uses index `i` and is responsible for the large pulses in the output. Once this loop terminates the second for loop is executed. This loop uses index `j` and produces the smaller pulses in the output. These two loops operate sequentially—one after another.

Part B

Now modify the code as follows: In the supplied code you will notice the line:

```
  };              // End of first FOR loop (Part A)
```

Change the line to look as follows (i.e. type in the two slashes):

```
//};              // End of first FOR loop (Part A)
```

This effectively removes the brace by "commenting it out" (everything from the first `//` to the end of the line is treated by the compiler as a comment). Further down you will see the following line:

```
//};              // New end of first FOR loop (Part B)
```

Remove the comment (the `//`) from the beginning of the line only, this will make the brace part of the code.

Compile and run the code. What does the output look like? How many pulses and what kind are they? *Again answer on answer sheet.*

The second loop (with index `j`) is now nested within the first loop (with index `i`). You can tell this by clicking behind the `{` on the first for loop (index `i`). BlueJ will highlight the braces (*i.e.*, `{` and `}`) entire piece of code that comprises the body of the for loop—the code that is controlled or affected by the loop.

Can you explain the difference in output between Part A and Part B?

Answer on answer sheet.

Part C

How many pulses, and of what sizes, would be produced if (in Part B) the limit in the outer loop (`i`) was changed to `6` and the limit on the inner loop (`j`) was changed to `3`? Why?

Again answer on answer sheet.

Part D

Open the `TwoByTwo` project in BlueJ. This program involves the use of two turtles (`yertle` and `mertle`) and two displayers (`display1` and `display2`). Run the program. A displayer will appear on the screen with an `OK` button on it. Drag the displayer to the left and you'll see a second displayer hidden underneath. Move the displayers so you can see all of both and then press the `OK` button. Drawing will now occur on the displayers. Does the program draw two squares on the two displayers? If not, can you explain why? Hint: Consider line 48 which says: `mertle = yertle;`

Answer on the answer sheet.