Activity 3.1.2

**Variables and Conditionals**

﻿Distance Learning

When complete, save this document and upload it to the appropriate Assignment page in **Canvas**.

Goals

* Apply coding fundamentals in a text-based language
* Develop and test code incrementally
* Develop a program independently

**After answering the nine Python Interpreter questions**, please submit a screen capture of the score and the first 1 or two questions here: (5 points)

Graphical user interface, text, application

Description automatically generated

**After Step 6**, please submit a screen capture of the Step 5 predictions and the embedded Trinket in Step 6 after running the code. Make sure that the screen capture contains the code and the output in the results window. (5 points)

Graphical user interface, application

Description automatically generated

**For Steps 8 through 10,** please add two additional print statements to the code to verify the data type of x and y. Run your code, type your input for name and age in the results window, and when the run is finished, make and submit a screen capture of the code and results window with the output. (10 points)

Graphical user interface, text, application, email

Description automatically generated

**For Steps 13 and 14,** make the additions to the code to convert the data type of the 5 values and print the results, and when the run is finished, make and submit a screen capture of the code and results window with the output. (5 points)

Graphical user interface, text, application

Description automatically generated

**After Steps 17,** Check your understanding of example arguments and return values, please submit a screen capture of the score and the first 1 or two questions here: (10 points)

Graphical user interface, text, application, email

Description automatically generated

**Step 18:** Why might it make sense to report the number of students in a class as an integer and not a float-point? (5 points)

|  |
| --- |
| It is because it is likely impossible to have a decimal of a student in the class |

**Step 19:** Three questions (15 points)

1. What do nested functions do?

|  |
| --- |
| It runs something, then it runs another part of the program |

2. How do nesting functions behave?

|  |
| --- |
| It runs whatever is the furthest within the function, then it keeps going out untill it reaches the end of the function. |

3. Which function executes first, and which last?

|  |
| --- |
| The one closest to the middle runs first, then the next one out, then it keeps going until it reaches the end of the function |

**Step 20:** After fixing the code, run the program, type in the appropriate input in the results screen, and when the run is finished, make and submit a screen capture of the code and results window with the output. (10 points)

Graphical user interface, application, Word

Description automatically generated

**For Steps 22 through 24:** Modify the 5 in each line of code so each statement prints True. Run the code and then create and submit a screen capture of the code and results window. (10 points)

Graphical user interface, application, Word

Description automatically generated

**After Step 24**, answer the 5 questions, “Determine whether each statement is True or False.” please submit a screen capture of the score and the first 1 or two questions here: (10 points)

Graphical user interface, text, application, email

Description automatically generated

**For Steps 29 through 31:** Modify the assigned value of the guesses variable until the result prints “You have run out of guesses!” Run the code and then create and submit a screen capture of the code and results window. (5 points)

Graphical user interface, text, application

Description automatically generated

|  |
| --- |
| **Please answer the following questions on this submission form: (10 points)** |
| 1. Give examples of code where you have seen data types and conditionals before.  These types of data types and conditionals are basically everywhere in coding except for a language like HTML. You can find all of these in JavaScript, Java, Kotlin, Go, PHP, Assembly, and even esoteric languages like Chef and Befunge  2. Why is Python considered to have a high level of abstraction even though you don’t see any blocks?  **Even though it is more complicated than something like blocks, Python is still very abstracted compared to languages like C and Assembly** |