

# Honors Computer Science Python

## Mr. Clausen

### Program 3A, 3B, 3C, 3D

#### Program 3A “Three For Time” (20 points)

Write a program to practice “for loops”. Ask the user to enter values for the lower case lower limit, lower case upper limit, upper case lower limit, upper case upper limit, and number upper limit in this order. Make your program user friendly by prompting them for this value.

To see a model for this program look at the source code for the program “taxform.py” in the network directory titled: “HnrCSPythonFiles”. Look in the folder “Student Data Files” and “Ch\_02\_Student\_Files”.

- 1) Use a `DocString` at the beginning of the program for your comments. Type comments at the beginning of the program to display your name and other information just like those used for program 1A. **Be sure to change the program name, program number, and program description.**
- 2) Leave a blank line after the `DocString`.
- 3) Type: **import math** (to import all of the math functions in the math module. Use the square root function in this program.)
- 4) Leave a blank line after the import statement.
- 5) Initialize all of the variables that are to be used in this program. Initialize each variable on a separate line. (Initialize integers to 0 (zero), decimal numbers to 0.0, strings to “” (double quotes with nothing in between them, and Boolean variables to False). You will need variables for all of the following: lower case lower limit, lower case upper limit, upper case lower limit, upper case upper limit, and number upper limit. Make sure that you use descriptive identifiers for all of your variables to model “self-documenting code”.
- 6) Leave a blank line after the variable initialization statements.
- 7) Type the following comment:  
#-----Display My Information-----  
Follow this comment with print statements to display your name and period output just like those used for program 1A. **Be sure to change the program name, and program number.**
- 8) Leave a blank line after the print statements listed above.

9) For the Input section, type the following comment:

```
#-----Input-----
```

Ask the user for the values of the lower case lower limit, lower case upper limit, upper case lower limit, upper case upper limit, and number upper limit. Be sure to type cast each of these inputs to “int” before you assign them to the variables.

10) For the Calculations and output section, type the following comment:

```
#-----Calculations & Output-----
```

Perform all of the calculations and output statements.

First, use a “for loop” to print out the capital letters of the alphabet from “A to Z” from left to right across the screen using their ASCII code values (65-90) in the loop and converting them to characters (See Chapter 2 for how to convert the ASCII code to the characters). Follow the example of the for loop “header” on slide 8 of the Chapter 3 PowerPoint presentation where they show how to use a lower limit and an upper limit. **This should only use 2 lines of code.**

Include 2 blank print statements to separate your capital letters from the lower case letters described below.

Second, use a “for loop” to print the lower case letters of the alphabet BACKWARDS from “z to a” from left to right across the screen using their ASCII code values (122-97) in the loop and converting them to characters. (See Chapter 2 for how to convert the ASCII code to the characters). Follow the example of the for loop “header” on slide 13 of the Chapter 3 PowerPoint presentation where they show how to count backwards in a “for loop”. **This should only use 2 lines of code.**

Include 2 blank print statements to separate your lower case letters from the table described below.

Third, use a for loop to print out the numbers from 1 to the number upper limit, their squares, cubes, square roots, and cube roots. Use format strings to display these titles first, then use format strings to display the numbers to align to the titles. Display square roots, and cube roots to 4 decimal places. Separate the results of the three “for loops” with blank lines. Follow the example of using format specifiers to display a table on slide 20 of the Chapter 3 PowerPoint presentation or page 89 of the textbook.

11) Finish your program with these last 2 lines of code.

```
print ("")  
input("Press enter to quit the program")
```

12) Save your program as LastNameFirstNameP3A.py.

13) When you are finished with your program, have tested it thoroughly to make sure that your program is correct, and are sure that you don’t need to make any changes, then save your program in the “T” network mapping, in the Program 3A folder.

**Program 3B** Tri-angles (20 points)

Write a program to practice “if statements with compound conditions”, “nested if statements”, and “if, elif statements” and Boolean variables. Ask the user to enter the values of three sides of a triangle. First determine if the three sides form a triangle (the sum of any two sides is greater than the third side – check all combinations). If the three sides form a triangle, check to see if it is a right or equilateral triangle. Tell the user if the sides form a triangle or not. If they form a triangle, tell the user if they form a right or equilateral triangle or other triangle.

Make your program user friendly by prompting them for these values. To see a model for this program look at the source code for the program “taxform.py” in the network directory titled: “HnrCSPythonFiles”. Look in the folder “Student Data Files” and “Ch\_02\_Student\_Files”.

- 1) Use a `DocString` at the beginning of the program for your comments. Type comments at the beginning of the program to display your name and other information just like those used for program 1A. **Be sure to change the program name, program number, and program description.**
- 2) Leave a blank line after the `DocString`.
- 3) Initialize all of the variables that are to be used in this program. Initialize each variable on a separate line. (Initialize integers to 0 (zero), decimal numbers to 0.0, strings to “” (double quotes with nothing in between them, and Boolean variables to False). You should have variables named `side1`, `side2`, `side3`, `isTriangle`, `isRightTriangle`, and `isEqualateralTriangle`, and any other variables that you wish to add. Make sure that you use descriptive identifiers for all of your variables to model “self-documenting code”, and that all variables are initialized at this place in the program.
- 4) Leave a blank line after the variable initialization statements.
- 5) Type the following comment:  
#-----Display My Information-----  
Follow this comment with print statements to display your name and period output just like those used for program 1A. **Be sure to change the program name, and program number.**
- 6) Leave a blank line after the print statements listed above.
- 7) For the Input section, type the following comment:  
#-----Input-----  
Ask the user for the lengths of the three sides of the triangle. Be sure to type cast each of these inputs to “int” before you assign them to the variables.
- 8) For the Calculations section, type the following comment:  
#-----Calculations-----  
These calculations are going to be a little different than just crunching numbers. Determine if the three sides form a triangle using an “if” statement with compound Boolean expressions. Assign True to the variable `isTriangle`, if the three sides form a triangle. If they do form a

triangle, determine if they form a right triangle (use the Pythagorean Theorem) using a nested “if” statement and a compound Boolean expression. If they form a right triangle, assign True to the variable isRightTriangle. Check to see if the sides form equilateral triangle using a compound Boolean expression in an “if” statement. Assign True to the variable isEquilateralTriangle, if the three sides form an equilateral triangle.

9) For the Output section, type the following comment:

```
#----- Output-----
```

Echo (print) the three sides and tell the user if they form a triangle, a right triangle, an equilateral triangle, or don’t form a triangle at all. Use nested “if” statements and “if elif” statements to print which type of triangle they are. Be sure to use the Boolean variables isTriangle, isRightTriangle, and isEqualateralTriangle in these “if” statements.

10) Finish your program with these last 2 lines of code.

```
print ("")  
input("Press enter to quit the program")
```

11) Save your program as LastNameFirstNameP3B.py.

12) When you are finished with your program, have tested it thoroughly to make sure that your program is correct, and are sure that you don’t need to make any changes, then save your program in the “T” network mapping, in the Program 3B folder.

### Program 3C “Get Random for a While” (20 points)

Write a program to practice “while loops”. Ask the user to enter values for the number of random numbers they want to generate, and the upper limit for the random numbers that they wish displayed. Make your program user friendly by prompting them for these values. Use one line comments to separate this program into its parts: input, and “calculations and output” which will be merged into one section. **DO NOT use “while True” style loops like the textbook!** To see a model for this program look at the source code for the program “taxform.py” in the network directory titled: “HnrCSPythonFiles”. Look in the folder “Student Data Files” and “Ch\_02\_Student\_Files”. Look in the Ch\_03\_Student\_Files folder for a program called, “checkinput using Primed While Page 107.py” for a model of a “primed while loop”.

- 1) Use a DocString at the beginning of the program for your comments. Type comments at the beginning of the program to display your name and other information just like those used for program 1A. **Be sure to change the program name, program number, and program description.**
- 2) Leave a blank line after the DocString.

- 3) import random.
- 4) Leave a blank line after the import statement.
- 5) Declare and initialize constants for the lower and upper limits of how many random numbers the user could ask to generate between 1 and 100,000. Declare and initialize constants for the lower and upper limits of what each random number could be between 1 and 100. Don't forget that constant names should use ALL\_CAPS.
- 6) Leave a blank line after the constants.
- 7) Initialize all of the variables that are to be used in this program. Initialize each variable on a separate line. (Initialize integers to 0 (zero), decimal numbers to 0.0, strings to "" (double quotes with nothing in between them, and Boolean variables to False). Make sure that you use descriptive identifiers for all of your variables to model "self-documenting code", and that all variables are initialized at this place in the program. **In the while loop that generates the random numbers, you will need to use a variable like "counter". Whatever you decide to name it, initialize it to 1 (one), not zero.**
- 8) Leave a blank line after the variable initialization statements.
- 9) Type the following comment:  
#-----Display My Information-----  
Follow this comment with print statements to display your name and period output just like those used for program 1A. **Be sure to change the program name, and program number.**
- 10) Leave a blank line after the print statements listed above.
- 13) For the Input section, type the following comment:  
#-----Input-----  
Ask the user for the number of random numbers they want to generate, and the upper limit for the random numbers that they wish displayed. Be sure to type cast each of these inputs to "int" before you assign them to the variables. Use "Primed While Loops" to check that the values are within the range of 1 to 100,000 for the number of random numbers displayed, and within the range of 1 to 100 for the upper limit of each random number. **Use the constants you declared as the lower and upper limits in your primed while loops.**
- 11) For the Calculations and output section, type the following comment:  
#-----Calculations & Output-----  
Perform all of the calculations and output. Use a "count controlled while loop" to display the random numbers with 10 numbers per row (an "if statement" will help accomplish this) and format strings so that all numbers are right justified with equal spacing between the numbers. (You should be able to do this in 6 lines of code.)

12) Finish your program with these last 2 lines of code.

```
print ("")
input("Press enter to quit the program")
```

13) Save your program as LastNameFirstNameP3C.py.

14) When you are finished with your program, have tested it thoroughly to make sure that your program is correct, and are sure that you don't need to make any changes, then save your program in the "T" network mapping, in the Program 3C folder.

### **Program 3D Choose Something From The Menu (20 points)**

Write a program to convert Celsius temperatures to Fahrenheit and vice versa. Display a menu and allow the user to choose which conversion they wish to do. Use a "while loop" Boolean condition similar to the sample code at the top of page 104 that will quit the program if the user presses the enter key without making a menu choice. Make your program user friendly by prompting them for the temperatures. **DO NOT use "while True" style loops like the textbook!**

- 1) Use a `DocString` at the beginning of the program for your comments. Type comments at the beginning of the program to display your name and other information just like those used for program 1A. **Be sure to change the program name, program number, and program description.**
- 2) Leave a blank line after the `DocString`.
- 3) Declare a constant: `FREEZING_POINT_OF_WATER= 32.0`.
- 4) Leave a blank line after the constants.
- 5) Initialize all of the variables that are to be used in this program including `celsius`, `fahrenheit`, and `menuChoice`. Initialize each variable on a separate line. (Initialize integers to 0 (zero), decimal numbers to 0.0, strings to "" (double quotes with nothing in between them, and Boolean variables to False). Make sure that you use descriptive identifiers for all of your variables to model "self-documenting code", and that all variables are initialized at this place in the program. **In this program initialize the variable `menuChoice` = "" to a space instead of an empty string.**
- 6) Leave a blank line after the variable initialization statements.
- 7) Type the following comment:  
#-----Display My Information-----  
Follow this comment with print statements to display your name and period output just like those used for program 1A. **Be sure to change the program name, and program number.**
- 8) Leave a blank line after the print statements listed above.

9) Start the while loop here. Look in the Ch\_03\_Student\_Files folder for a program called, “sum.py” to see a model of a while loop that will exit the loop when the user presses the enter key without pressing one of the menu choices to quit.

10) Inside the while loop, display a menu that looks like the following using print statements:

Main Menu: Temperature Conversion

1. Convert Celsius to Fahrenheit

2. Convert Fahrenheit to Celsius

Press enter without choosing 1 or 2 to quit

11) Ask the user to enter their choice and DO NOT type cast what they type. We need to keep that response as a string variable.

12) Use “if, elif statements” to decide which conversion to perform depending on whether the user entered a 1 or a 2. In each of the “if choices” ask the user to enter the temperature (type cast this to “float”, perform the calculations for the temperature conversions, and display the converted temperature. Here are the formulas:

$$\text{celsius} = (\text{fahrenheit} - \text{FREEZING\_POINT\_OF\_WATER}) * 5.0 / 9.0$$
$$\text{fahrenheit} = 1.8 * \text{celsius} + \text{FREEZING\_POINT\_OF\_WATER}$$

13) Finish your program with these last 2 lines of code.

```
print ("")
```

```
input("Press enter to quit the program")
```

14) Save your program as LastNameFirstNameP3D.py.

15) When you are finished with your program, have tested it thoroughly to make sure that your program is correct, and are sure that you don't need to make any changes, then save your program in the “T” network mapping, in the Program 3D folder.