Introduction to Computer Science (ICS 3U)
User Defined Single Result (Pure) Function Assignment

Create a folder called **User Function** and save the following programs in that folder. You should also create a folder called **Useful Functions** to store useful user-defined functions you create. Note the functions created in this section must have at least one result command and should not contain the input () or print () functions.

1. Create a single result function that will accept two integers and will display the difference between them. (ie. The difference between -2 and 5 is 7). Note that a difference cannot be a negative value. Use the function in an appropriate program at least twice. Call your program **UserFunction1.py**

2. Create a single result function that will convert temperature in degrees Fahrenheit to degrees Celsius. Create a 2nd function that will convert temperature in degrees Celsius to degrees Fahrenheit. Use your functions in an appropriate program. Call your program **UserFunction2.py**

   Notes: - To convert Celsius to Fahrenheit, use the following equation: 9 / 5°C + 32
   - To convert Fahrenheit to Celsius, use the following equation: (°F - 32) * 5 / 9

3. Create a single result function that will accept a mark and return an appropriate letter grade (ie. A, B, C, etc.). Use your functions in an appropriate program. Call your program **UserFunction3.py**

4. Create **four** different single result functions that will accept three string parameters which are the person's first, middle and last name.
   - The first single result function will return the initials of the person separated by a period (ie. John Bobby Smith will have J.B.S. returned).
   - The second single result function will return the name all capitalized (ie. John Bobby Smith will have JOHN BOBBY SMITH returned).
   - The third single result function will return the eLearning userid which is the first letter of the first name, the first initial of the middle name and the full last name (ie. John Bobby Smith will have JBSmith returned).
   - The fourth single result function will return the school network userid which is the first four letters of the first name and the first four letters of the last name with "22" at the end (ie. John Bobby Smith will have JohnSmit22 returned). Assume that all first and last names have at least 4 letters. *Level 4 - Make the function work for any names, even if there are fewer than 4 letters.*

   Use each of your functions at least once in an appropriate program. Call your program **UserFunction4.py**

5. Python is an unusual programming language because the type of variable is determined when a value is assigned to the variable. In most programming languages (like Turing), the type of variable must be determined when the variable is declared. This characteristic of Python can be used to create single functions which work differently according to the type of data that is passed in as parameters.

   Create a single result function that will accept three parameters. It will add the values if the parameters are integers or float values and will concatenate the values if the parameters are strings and return the value. Note that this simpler than it at first appears!

   Use your function twice in a program which asks the user to input 3 numbers to be added and then asks the user to input three strings to be concatenated. Call your program **UserFunction5.py**

6. Create four single result functions which will accept a single parameter which is a string with only letters (no numbers or special symbols).
   - The first single result function will return the number of words in the string (hint: you can find the number of words by finding the number of spaces)
   - The second single result function will return the number of characters (do not include any spaces in the count)
   - The third single result function will return the number of vowels
   - The fourth single result function will return the number of consonants (hint: use the 2nd and 3rd functions in this function).

   Use your functions at least once in an appropriate program. Call your program **UserFunction6.py**

7. Lotto 649 is a lottery that allows the user to choose any six numbers between 1 and 49 (ex. 3, 6, 23, 31, 40, 48). Create a single result function which will produce a random Lotto 649 number. This function will not have any parameters but will return a string containing the six numbers, each separated by a space. Your program using the function should ask the user how many numbers they want and call the function the required number of times. Call your program **UserFunction7.py**

   Note: Even if a function has no parameters, an empty parameter list must be included in the function definition and function call.

8. An online store owner has come to you for help. They sell most of their goods in Ontario but sell some items to customers in other provinces and the states. They need a function to calculate the total price which tax. In Ontario, the harmonized tax rate is 13% but it is different in other areas. To make their life easier, create a function which will accept only the price if they sell to a customer from Ontario. If they sell to a customer from outside Ontario, the function will accept the price **and** the tax rate of that state or province. Call your program **UserFunction8.py**

   Hint: Use a default parameter (argument) for tax rate.

9. Leap years occur according to the following formula: a leap year is divisible by four but not by 100 unless it is divisible by 400. For example, 1992, 1996 and 2000 are leap years but 1900 and 1993 are not. The next leap year that falls on a century will be 2400. Create a function that will accept a year and return an appropriate boolean value (True or False) if the year is or isn't a leap year. Use your functions in an appropriate program. Call your program **UserFunction9.py**

10. a) Create a function which accepts two real parameters - the interest rate and the principle of a loan in dollars. It should then return the dollar amount of interest rounded to two decimal places. Call your program **UserFunction10.py**

    b) Improve the function you used in (a) so that it returns the number with dollar signs. Re-save your program as **UserFunction10.py**

    c) Improve the function you used in (b) so that there are no 0's dropped. (ie. When the number 9.99 is rounded, 10 will appear. Make the program output $10.00.) Re-save your program as **UserFunction10.py**