COMP1005/1405 Fall 2024 Tutorial 4

Do not post the tutorial or the solutions on any website.

Objectives

- Practice writing/running Python code in VSCode
- Practice creating **circuit diagrams** and **flowcharts**
- Practice applying **branching and looping structures** in Python programs

Expectations

To receive full grades for this tutorial, you must complete Problems 1-3.

Grading Scheme for Tutorials

For each tutorial, you will be graded based on the following criteria:

- 2/2 for demonstrating problems 1-3 and providing **YourStudentNumber_T4.zip** file
    - Section A Tutorial Sessions: submit your work to the tutorial BrightSpace. The zip file should contain your solutions to all the required problems.
    - Section C, D, and E Tutorial Sessions: demonstrate your tutorial work in person to a teaching assistant by the end of the tutorial session.
- 1/2 if you are missing problems or your solutions need significant improvement.
- 0/2 if you do not submit to BrightSpace or demonstrate to a teaching assistant
    - Section A Tutorial Sessions: no submission to the tutorial BrightSpace.
    - Section C, D, and E Tutorial Sessions: not demonstrating your tutorial work to a teaching assistant by the end of the tutorial session or being unable to answer the teaching assistant's questions about your tutorial.

Problem 1 (Diagrams)

Attach all your diagrams to a document and save it as a PDF file named **diagrams.pdf**. **Note you can draw the circuit diagrams and flowcharts on paper and then scan/ photograph your work to be added to the document.**

1. Circuit Diagram

a. ((A or B) and (not C)) and (not D)
b. X or (not (Y and Z))

2. Flowchart

a. Draw a flowchart representing a program that accepts an integer input from the user and determines whether the number is even or odd.
   Hint: Use conditional operators and math operators.

b. Draw a flowchart representing a program that accepts an integer input from the user and determines whether the number is less than zero, equal to zero or greater than zero.
   Hint: **Think of nested conditional statements.**

Problem 2 (Branching)

1. Write a Python file named **`oddEven.py`** in VSCode to implement the Even or Odd Number validation program. Convert your flowchart of problem 1.2.a into a working Python program.

2. Write a Python file named **`threePaths.py`** in VSCode to implement the nested branching program. Convert your flowchart of problem 1.2.b into a working Python program.

Problem 3 (Loop)

Write a Python file named **`guessing.py`** in VSCode to implement the Guessing Game program.

1. To begin your program, you must generate a pseudo-random integer between 1 and 100 (inclusive).

To do this, include the following line at the top of your Python file:

```
import random
```

Note: This statement imports the random library, enabling you to use its pseudorandom number generator.

To generate a random integer, use the randint() function, which **returns a random integer** between the start and end values **inclusively**.

```
random.randint(start, end)
```

For example, random.randint(3,4) will return either 3 or 4. For more information, you can refer to the randint() documentation. We will learn more about libraries and modules later in the course.

2. The user will have a maximum of 10 attempts to guess the selected number.
3. After each incorrect guess, your program should indicate whether the actual number is higher or lower than the randomly chosen number.
4. If the user guesses correctly: "You are correct the number was X." and terminate the guessing game loop; if the user runs out of guesses: "The correct number was X."

Sample Output (suggested output):

```
Welcome to the guessing game! You have up to 10 attempts to guess the number.
Attempt #1: Guess a number between 1 to 100: 50
        Your guess was too high.
Attempt #2: Guess a number between 1 to 100: 20
        Your guess was too low.
Attempt #3: Guess a number between 1 to 100: 40
        Your guess was too low.
Attempt #4: Guess a number between 1 to 100: 45
        Your guess was too high.
Attempt #5: Guess a number between 1 to 100: 43
        Your guess was too high.
Attempt #6: Guess a number between 1 to 100: 44
        Your guess was too high.
Attempt #7: Guess a number between 1 to 100: 42
        Your guess was too high.
Attempt #8: Guess a number between 1 to 100: 41
        You are correct the number was 41
```

Compress Files (zip files)

1. Create a directory/folder and copy/move all your tutorial files to it.
2. Enter this directory, select all files, and create a zip file as described above. Name it `YourStudentNumber_T4` (replace **YourStudentNumber** with your 9-digit student number).

Final Step

For Section A (Submit the work before the tutorial ends):

1. **Submit** your **zip** file to our Merged Tutorial Brightspace. The due date of your submission is aligned with your tutorial session.
2. **After** you submit the file, download your submission from Brightspace and confirm that it is a zip file containing **diagrams.pdf**, **oddEven.py, threePaths.py** and **guessing.py**. **Extract** the .py files and execute those again to ensure they work properly. Occasionally, a problem can occur during the upload process, and files can become corrupted.

For Sections C, D and E (Show the TAs your work before the tutorial ends):

1. **Problem 1:** Show pdf file to TAs and explain your diagrams.
2. **Problem 2-3:** Run your Python programs in VS Code to demonstrate they are working.
3. Answer the questions TA may ask.
4. Show the TA your zip file and extract the files.