

Specification for Assignment 2 for COMP1405 (Fall 2016)**Instructions**

There are two questions in this assignment. For each question you will be submitting your solution as multiple files, each containing source code written in Python 3. These files must be compressed into a single "zip" file, and you will submit this file using cuLearn.

- ☐ *The source files for question 1 must be named "a2q1a.py", "a2q1b.py", and "a2q1c.py"*
- ☐ *The source files for question 2 must be named "a2q2a.py" and "a2q2b.py"*
- ☐ *The compressed archive (i.e., the .zip file) must be named "a2.zip".*
- ☐ *The due date for this assignment is October 8, 2016, by 11:30pm.*



Late assignments will be accepted for 48 hours after the deadline, but the penalty for submitting a late assignment is a loss of 2.0% per hour.



You are expected to demonstrate good programming practices at all times (e.g., choosing descriptive variable names, provide comments in your code, etc.) and your code will be penalized if it is poorly written.



You are expected to do the necessary preparatory work (e.g., devising an algorithm) before you begin coding. Whenever appropriate, you will be asked to present either pseudocode or a flowchart before you will receive any assistance from the instructor or a teaching assistant.



This assignment is uniquely generated; every student in the class is required to complete a slightly different version of this assignment. To ensure that each unique assignment shares the same level of difficulty, a unique assignment generator (and supporting files) has been made available on cuLearn.

To receive the assignment instructions that are specific to you, download the "unique-assignment-generator-for-A2.zip" file from cuLearn. Once you have extracted the contents to your working folder, use the command prompt to run the "generator-for-A2.py" program and then enter and confirm your student number.

Specification for Assignment 1 for COMP1405 (Fall 2016)

Question 1 (of 2)

When you ran the unique assignment generator for this assignment, the program showed you a collection of 18 cartoon faces as the top three rows of the SimpleGraphics window. For the first question of this assignment you will create three short programs that will ask the user questions as though the user is playing a game of "Guess Who?" using this collection of faces. If you are completely unfamiliar with this game you should consider visiting the Wikipedia page (https://en.wikipedia.org/wiki/Guess_Who%3F) and playing an online implementation (such as the one at <http://www.miniplay.gr/?view=game&gid=76>).

Each of the programs you write will require you to use the input function to ask the user one or more yes or no questions. You **MUST** restrict your questions to asking about whether or not the player's choice of face has a **wig**, **glasses**, **hat**, **pipe**, **moustache**, **beard**, or **scar**, and each question can only ask about one of those features. To clarify, you are **NOT** permitted to ask questions pertaining to the row, column, eyes, mouth, or any feature other than the seven listed above.

Your first program (i.e., "*a2q1a.py*") must contain a single if statement that does not use any of the Boolean operators (i.e., and, or, not), has no elif or else branches, and would only work to identify one specific face (i.e., ending the game with a single question). To clarify, you can assume that your program will win if the opponent (i.e., the user) has chosen the only face out of group of eighteen that can be identified by a single question. For further clarification, consider the collection of three faces below and note that the only single question that could conceivably identify one of these three is the question "Are you wearing a hat?"



Your second program (i.e., "*a2q1b.py*") must contain a single if-else statement that does not use any of the Boolean operators (i.e., and, or, not), has only a single else branch, and is guaranteed **NOT** to identify your opponent. To clarify, regardless of which face your opponent has chosen, they should end up following the else branch of your statement.

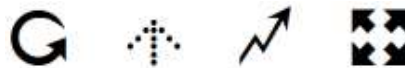
Your third program (i.e., "*a2q1c.py*") must contain a single nested if statement with a depth of two (i.e., one "inner" if statement nested inside one "outer" if statement) that does not use any of the Boolean operators (i.e., and, or, not). Write this program such that exactly five possible choices of face should pass the condition of the outer if statement, and only one of those faces should pass through the condition of the inner if statement.

Specification for Assignment 1 for COMP1405 (Fall 2016)

Question 2 (of 2)

When you ran the unique assignment generator for this assignment, the program also showed you a collection of 10 arrows as the bottom two rows of the SimpleGraphics window. For the second question of this assignment you will create two expert systems that will allow the user to uniquely identify each of these arrows by answering a series of questions.

An expert system represents to a solution to a categorization problem. An expert system is expected to maintain a database that contains the identity of each element of a large collection, and then when a user has an unknown element they would like to identify they can answer questions from the expert system and the expert system can identify the unknown element. As a clarifying example, suppose the database maintained by an expert system contained the following four arrows (denoted arrows 1, 2, 3, and 4, respectively)...



...and the user encountered the following unknown symbol, to be identified:



The expert system might ask the user "Does the symbol use any curved lines?" and if the user's answer is "yes" then the expert system knows the unknown symbol must be either 1 or 2. The expert system might then ask "Does the symbol point straight upwards?", etc.

Your first expert system (i.e., "*a2q2a.py*") must use a single if-elif-else statement to correctly identify each of your ten symbols, and this statement must be flat (i.e., not nested). For this question you are only permitted to ask the user yes or no questions. To accomplish this you will need to use Boolean operators (i.e., and, or, not) and create conditions such that each condition only evaluates to True for a single arrow. As a clarifying example you might create the following flat if statement (for the example included above):

```
if isCurved == "yes" and isUpwards == "yes":  
    print(2)  
elif isCurved == "yes" and not(isUpwards == "yes"):  
    print(1)  
...
```

Your second expert system (i.e., "*a2q2b.py*") must use a single nested if statement to correctly identify each of the ten symbols that you have been assigned. This system is expected to ask as many non-yes or no questions as possible (e.g., "How many solid parts make up the symbol?") but it must not ask any unnecessary questions. This entails that, unlike your first expert system, your second system cannot ask all the questions up front - some of your input function calls must be made inside the nested loop.