COMP1005/1405 – Winter 2022
Assignment # 5
Due on Friday, April 8th by 11:59 pm.

Submission Guidelines:

Submit a single zip file called A5.zip on Brightspace.

Did you zip it correctly? You must always create the zip files using the built-in, default, archiving program available with your operating system. If we cannot easily open your zip file and extract the python files from them, we cannot grade your assignment. Other file formats, such as rar, 7zip, etc., will not be accepted.

Windows: Highlight (select with ctrl-click) all of your files for submission. Right-click and select “Send to” and then “Compressed (zipped) folder”. Change the name of the new folder “A5.zip”.

MacOS: Highlight (select with shift-click) all of your files for submission in Finder. Right-click on one of the files and select “compress N items…” where N is the number of files you have selected. Rename the “Archive.zip” file “A5.zip”.

Linux: use the zip program.

Did you submit the correct files? You are responsible for ensuring that you have submitted the correct file(s), and the submission is completed successfully. After submitting your A5.zip file to Brightspace, be sure that you download it and then unzip it to make sure that you submitted the correct files. This also checks that your zip file is not corrupted and can be unzipped.

You can submit as many times as you wish and Brightspace will save your latest submission. I would suggest that you submit as soon as you have one question done and keep re-submitting each time you add another problem (or partial problem).

We will not accept excuses like “I accidentally submitted the wrong files” or “I didn’t know that the zip file was corrupt” or “My Internet was down” or “My computer was not working” after the due date.

If you are having issues with submission, contact the instructor before the due date.

Late Policy

The assignment is due on Friday, April 8th, 2022 by 11:59 PM (Ottawa time). Late submissions are allowed until Sunday, April 10th, 2022, 11:59 PM (Ottawa time) without any penalty.

Multiple submissions are allowed and Brightspace will only keep your LAST submission. The submission link will close after Sunday, April 10th, 2022.

Here is a summary of the penalties based on the submission time.

Last submission Day/Time	**Penalty applied**
Friday, April 8th, 2022 by 11:59 PM (Ottawa time)	No penalty
Sunday, April 10th, 2022, 11:59 PM (Ottawa time)	No penalty
Monday, April 11th or later	100%

Regret Clause

If you think you have committed an academic violation (plagiarism), you have 12 hours after the final allowable submission time (12 pm on Monday, April 11th, for this assignment) to invoke this regret clause.

If you do, you will receive ZERO for the assignment (or portions of it that you invoke the clause for) but will not receive any further sanctions from the Dean's office. Your assignment will still be used when we are checking for plagiarism, but you would not face any further sanctions if it was decided that you had committed an academic violation.

For example, if you collaborated too closely with another student and this is discovered, you would not be further sanctioned. The other student, if they did not also invoke this clause, may still be further sanctioned by the Dean's office.

This is not a group project, so collaboration is not allowed on this assignment.

Self Evaluation Quiz Criteria

SE Quiz 5 is a follow-up quiz for Assignment-5. When you will submit this assignment, you can take the quiz on Brightspace.

You will be asked some leading questions to help you assess the problem-solving skills that you might have achieved by doing this assignment, such as:

- Do you feel more comfortable with programming terminologies?
- Did you draw a flowchart or write a pseudocode to help understand an algorithm?
- Did you try to break your problem into smaller sub-problems and solve each sub-problem separately?
- Did you use examples to help understand a problem?
- Could you explain the problems in your own words to someone else in this class?
- Did you find similar or related statements that we've talked about in lecture or tutorial, and did you try to understand how the problems were similar and how they were different?
- Did you ask a question on discord that required you to communicate what you understood and what was still confusing?
- Did you answer a peer's question on discord?
- Do you feel more comfortable using Python data structures like lists and dictionaries?
- Do you feel more comfortable using strings and File I/O methods to read from, process, write to files in Python?
- Did you trace your code/algorithm and make sure it follows correctly and produce correct results?
- Did you try to identify the stages of the programming life cycle while you were solving a problem for this assignment?
- Do you feel more comfortable with identifying the errors in your code and handling those properly?
- For the given assignment, which letter grade would you give yourself (A+, …, F)?
- In 2-3 sentences, justify your letter grade selection.

Note: only the last two questions are mandatory for this quiz.

Topics: Functions, Data Structures, Exception handling, OOP

In this last assignment, you will apply the object-oriented approach to the game you previously wrote for A2. The output of the game will remain the same but you will re-design the code so that it must use the following programming tools.

- classes and objects to represent the elements of this game
- data structures to store and manage data efficiently
- exception handling to catch and handle runtime errors,
- functions to divide a large problem into smaller ones and to reuse codes where necessary

You may want to use the algorithm that you developed for A2 since the conditions of the game have not changed. This assignment aims to produce a cleaner, modularized and well-designed code.

This specification provides the requirements that must be used to re-design the code for A5. Section 1 describes the classes that you should use to design the new data types. You may create additional classes and/or use additional attributes/methods for these classes. Section 2 guides you through the decomposition step to divide the game logic into smaller problems. You are required to write functions for those smaller problems (basically re-use your previous code and put them inside those functions) and you are encouraged to use additional functions where applicable. You can select any data structure(s) that you want to use in this assignment.

You will create and submit one py file **a5.py** for this assignment. Please be sure to read through the entire assignment before you begin to redesign your algorithm and start to code it. You may have to spend more time redesigning your algorithm than coding it. You are required to present your algorithm if you seek any help from the TAs or the instructor.

1. Required classes

Your program must implement the class definitions described in this section and create/use objects of these classes and methods where applicable. You can write additional classes/attributes/methods to your program. You must include the **`__init__`** method for these classes to initialize their attributes.

(a) **Door class** has at least three attributes: door number (int), treasure value (int/float), and selected (bool). The 'selected' attribute is initially 'False'. If a player selects this door at any level, this attribute changes to True for that level. You may want to write a method for this process.

(b) **Token class** has at least two attributes: name (string) of this token, i,e., 'silver'/'gold'/'diamond' and a list of the door objects (or the door numbers) that this token can open. This class also has a method that takes a door object (or a door number) and returns

True if this token can open that door, Otherwise, it returns False. It will be useful in the game to determine if the player's token can open a particular door.

(c) **Game class** has at least three attributes: the name (string) of the player, a list of values (int/float) won/lost by the player at each level, and a bool attribute named finished, which is initially False and only changes to True when the player decides to quit the game or completes all three levels of the game. This will be useful to determine when to end the loop/game.

Note that, it is possible to improve these class definitions, for example, you may want to create a separate class called Player, etc. You can add additional attributes and methods to these classes if necessary. You are welcome to discuss your plan with the instructor/TAs. Marks will not be deducted if you decide not to make any improvement and just use these classes.

2. Decomposing the problem and the required functions

To help you with the *decomposition* step, the game logic is broken down into five (5) smaller problems as follows.

1. Welcome the player, ask for the name of the player and show them the rules.
2. Display the options for choosing a door along with the values of the hidden treasure and the minimum token type to open each door.
3. Ask the user to select a door and if they want to change it. Perform the required input validation.
4. Allow the player to get a token and determine what the token type is.
5. Determine if the player wins/loses this level and store the amounts won/lost at this level.

To make the grading process consistent, the names and the tasks of these functions have been specified in this section. You can decide the parameters and the return values for each of these functions. You may want to divide the amount of work done in each of these functions into multiple smaller sections and write additional functions for them. Be sure to use them properly inside of the mandatory functions so they can perform the required tasks. Only the required functions will be graded.

Note that I have used the sample outputs from A2 so that you can refer to your previous algorithm/code and correctly decompose that for A5. If you previously customized your output in some way, you can follow that version.

1. Write a function named `intro()` that performs the task for the subproblem (1).

The sample output for this function can be:

```
Hey gamer, what's your preferred name?
>> Dark Horse
```

```
Welcome to the game, Dark Horse! Want to find some secret treasures? Let's begin.

You can choose a door to get the treasure hidden behind it.
But you must also collect a token from the magic wheel to open that door.
Check how much you can win and which token you need to open a door.
```

2. Write a function named `show_menu()` that performs the task for the subproblem (2).

   The sample output for this function can be the following:

   Output – 1:

```
Level-1
If you win this level, you'll get 1 x the value of the treasure.
But if you lose, 1 x the value of the treasure will be taken away from you.
----------------------------------------------------------------------
Door                    Value            Token type
[1] Door-1              $100.00          Silver
[2] Door-2              $200.00          Gold
[3] Door-3              $300.00          Diamond
```

   Output – 2:

```
Level-3
If you win this level, you'll get 3 x the value of the treasure.
But if you lose, 3 x the value of the treasure will be taken away from you.
----------------------------------------------------------------------
Door                    Value            Token type
[1] Door-1              $300.00          Silver
[2] Door-2              $600.00          Gold
[3] Door-3              $900.00          Diamond
```

3. Write a function named `select_door()` that performs the task for the subproblem (3).
   The sample output for this function can be the following:

```
Which door do you want to choose? Type [1, 2, or 3]. To quit the game enter 'q'.
>> -5
Invalid input, please enter a single digit between 1 and 3.

Which door do you want to choose? Type [1, 2, or 3]. To quit the game enter 'q'.
>> 5
Invalid input, please try again.

Which door do you want to choose? Type [1, 2, or 3]. To quit the game enter 'q'.
>> 1
Do you want to change your decision and choose another door? Enter 'Yes'/'No'.
>> maybe
Invalid input, please try again.
```

```
Do you want to change your decision and choose another door? Enter 'Yes'/'No'.
>> yes
Which door do you want to choose? Type [1, 2, or 3] >> 2

Dark Horse, you have selected Door-2. Please spin the magic wheel to collect your
token.
```

4. Write a function named `get_token()` that performs the task for the subproblem (4). This function does not print anything, so there is no sample output for it. It returns a token object.

5. Write a function named `update_amount()` that performs the task for the subproblem (5). This function prints something like the following and updates the list of values of the game object.

   Sample output when the player wins a level:

   ```
   Congrats!! You've won the hidden treasure worth $200.00.
   Your current total is $200
   ```

   Sample output when the player loses a level:

   ```
   Bad luck.. can't open Door-3 with a silver token, you lost $600.
   Your current total is $-400
   ```

   Alternatively, you can print this from the main() function.

6. Write a main function that must initialize all the necessary variables, create the required objects, and call all other functions in the correct sequence to play the game. Do not forget to add the main guard to call `main()`. Displaying the game summary is optional now. A complete sample run of this program is available in section 4.

3. Function definitions and parameters/return values

- Note that, for each function, you can decide the input parameters and the return values (if any).

- You must not use any global variables in this program but you can use global constants if needed. Recall that a global variable is declared outside of any function on the top level of the program that is available to the entire program and its value can be changed. And global variables should always be avoided if possible. Whereas, a global constant is also declared outside of any function on the top level of the program that is available to the entire program but its value must not be changed.

You can use this simple rule to determine the parameters/return values for any function. If you want to access some data (variable/data structure/object) inside of a function, instead of using a global variable you must pass that as a parameter to the function. Similarly, if your function creates or updates some value(s) that you want to make available to some other parts of your program, your function should return that value(s). In this way, the calling function should be able to use that value.

- Make sure that your program does not crash on valid inputs (valid inputs will not cause runtime errors, so it is not about exception handling). Otherwise, the TAs will have the discretion to decide if they want to debug your code and continue grading, or just give zero to the rest of your assignment.

- Marks distribution will be based on both the correctness of the output and the efficient use of the programming tools. So, if your program produces the correct outputs but does not use any of the required programming tools (that means it essentially is Assignment-2), it will not receive more than 40% of the total marks.

- You must add a brief docstring to each of your functions so that the grader knows how your functions should behave.

- Your functions must use exception handling to handle all possible run-time errors so they do not crash on invalid inputs.

4. A complete sample output

A sample run of your a5.py may look like this.

```
Hey gamer, what's your preferred name?
>> Dark Horse

Welcome to the game, Dark Horse! Want to find some secret treasures? Let's begin.

You can choose a door to get the treasures hidden behind it.
But you must also collect a token from the magic wheel to open that door.
Check how much you can win and the token you need to open a door.

Level-1
If you win this level, you'll get 1 x the value of the treasure.
But if you lose, 1 x the value of the treasure will be taken away from you.
----------------------------------------------------------------------
Door                    Value           Token type
[1] Door-1              $100.00         Silver
[2] Door-2              $200.00         Gold
[3] Door-3              $300.00         Diamond
```

```
Which door do you want to choose? Type [1, 2, or 3]. To quit the game enter 'q'.
>> 1
Do you want to change your decision and choose another door? Enter 'Yes'/'No'.
>> yes
Which door do you want to choose? Type [1, 2, or 3] >> 2

Dark Horse, you have selected Door-2. Please spin the magic wheel to collect your
token.
...... (drum roll) .... you've got a gold token (85.1%).


Congrats!! You've won the hidden treasure worth $200.00.
Your current total is $200

Do you want to play the next level (Level-2) (enter 'Yes'/'No')? >> yes

Level-2
If you win this level, you'll get 2 x the value of the treasure.
But if you lose, 2 x the value of the treasure will be taken away from you.
----------------------------------------------------------------------
Door                    Value            Token type
[1] Door-1              $200.00          Silver
[2] Door-2              $400.00          Gold
[3] Door-3              $600.00          Diamond


Which door do you want to choose? Type [1, 2, or 3]. To quit the game enter 'q'.
>> 3
Do you want to change your decision and choose another door? Enter 'Yes'/'No'.
>> no

Dark Horse, you have selected Door-3. Please spin the magic wheel to collect your
token.
...... (drum roll) .... you've got a silver token (37.1%).


Bad luck.. can't open Door-3 with a silver token, you lost $600.
Your current total is $-400

Do you want to play the next level (Level-3) (enter 'Yes'/'No')? >> yes

Level-3
If you win this level, you'll get 3 x the value of the treasure.
But if you lose, 3 x the value of the treasure will be taken away from you.
----------------------------------------------------------------------
Door                    Value            Token type
[1] Door-1              $300.00          Silver
[2] Door-2              $600.00          Gold
[3] Door-3              $900.00          Diamond


Which door do you want to choose? Type [1, 2, or 3]. To quit the game enter 'q'.
>> 3
```

```
Do you want to change your decision and choose another door? Enter 'Yes'/'No'.
>> yes
Which door do you want to choose? Type [1, 2, or 3] >> 1

Dark Horse, you have selected Door-1. Please spin the magic wheel to collect your
token.
...... (drum roll) .... you've got a gold token (87.5%).


Congrats!! You've won the hidden treasure worth $300.00.
Your current total is $-100
Thanks for playing!
```

Invalid submissions that may incur penalties

- Submissions with an incorrect file name or format
- Missing name and ID
- Submissions that crash (i.e., terminate with an error) on execution may receive a mark of zero (0).
- Others: functions without docstrings, use of global variables, missing required components, not using loops to repeat prompts and/or the levels of the game.

A5.zip Submission Recap

You must submit the python script **a5.py** with the A5.zip file.

Marking scheme

The marking scheme will be posted later on discord.