Specification for Assignment 1 for 101388288

The due date for this assignment will be **Friday, September 12, 2025, at 11:59pm EST**.

Late submissions are accepted without penalty until the following Sunday at 11:59pm EST.
Submissions received after that Sunday "cut-off" will not be accepted and will receive a mark of zero.

This semester, the offerings of "Introduction to Computer Science I" (i.e., COMP 1405 and COMP 1005) delivered by Dr. Robert Collier use techniques (such as procedural generation) to diversify the assessment activities, and the suite of six assignments that you will complete will be combined into a "portfolio game" project that you can use to showcase the skills that you have developed. This activity was inspired by a classic educational game known as "The Oregon Trail", originally developed as an educational tool to teach students about the hardships of pioneer life. The player would assume the role of a wagon leader and be tasked with crucial decisions (e.g., when to rest, how to cross rivers, etc.).

If you have never played "The Oregon Trail", you are required to do so before beginning this assignment.
It can be played online for free at https://oregontrail.ws/games/the-oregon-trail/play/.

The "theme" selected to form the basis for your portfolio game is **"Vendor Circuit"**

One of the objectives of your game will be to educate players about **"Food Preparation"**.

Please note that neither of these phrases can be used as the title for your portfolio game.

With this first assignment, you will begin laying the foundation for your project by showing a title screen, displaying some introductory text, receiving (and storing) names provided by the user as input, and practicing with the pipeline design pattern to allow the user to purchase an initial loadout of supplies. More specifically, after completing this assignment you will have demonstrated that you can:

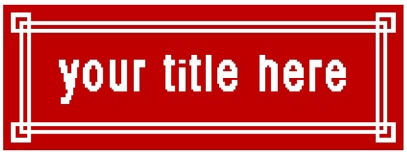
- ☐ install a library (specifically pygame; https://www.pygame.org/wiki/gettingstarted)
- ☐ implement a Python program that is well-commented and uses a "main" function
- ☐ use a command-line interface to navigate towards and execute a Python program
- ☐ import a module (that you will download via Brightspace) into a Python program
- ☐ read some code documentation (generated specifically for the module provided)
- ☐ explain the string data type and print text to the terminal using f-strings
- ☐ receive user input from the command-line and store it using simple variables
- ☐ change inputs into a numeric type and implement an arithmetic "pipeline"
- ☐ take a screenshot and compress your submission files into a ZIP archive
- ☐ complete a Brightspace "quiz" activity (to register your project details)

The library of functions created specifically for you for this assignment is available on Brightspace as **"portfolio_game_base_module_for_101388288.py"**.

Please note that this is the only import file that is permitted for the first assignment. If you make any additional imports then you will receive a grade of zero for the first assignment.

Since this is only the first of the specifications for the portfolio game project, a detailed procedure for completing this assignment has been provided below. Please also note that your submission must not use any branching or repeating control structures, nor is it permitted to use collections (e.g., lists) or include functions that were not specified below. Do not import a module other than **"portfolio_game_base_module_for_101388288.py"**.

1. Start by completing the "Rough Work for Assignment 1" activity on Brightspace to register key elements of your submission. You can complete this activity as many times as you wish before the cut-off date and only the latest will be considered.
2. Before beginning your implementation, create a working folder in a convenient location, download the **"portfolio_game_base_module_for_101388288.py"** module, move it into this folder, and then create a text file for your source code, giving it the file name of **"comp1405_f25_101388288_assignment_01.py"**. Don't forget that, when importing this file, you must not include the '.py' extension in the import statement.
3. At the beginning of your source file you must include your name and student number as "comments", and then once you have imported the base module that has been provided, create a definition for your "main" function (and include a call to this function at the bottom of the file).
4. Your first implementation task will be displaying a simple title screen. Once you have fully read the documentation provided, you should start by making a function call to open a new display window that is 640 pixels wide and 480 pixels high, making another function call a function to change the background of this window to the colour known as "**storm**", and make another function call to display your game title, centered horizontally at a vertical position of **180**, in a colour of your choice. Consider selecting a colour that contrasts well with the background.
5. Make several function calls (such as those for drawing lines, rectangles, and/or polygons) to create a tight border around your title text. You may use any colour you wish, but the border you create must copy the style that is depicted below. It is strongly recommended that you use graph paper to help you work out the co-ordinates and dimensions.



6. Add another function call to keep the window open for **4** seconds or until the user presses a key, and once the title screen has closed and focus has shifted back to the command prompt, your program must print some introductory information (i.e., introduce your game) on the terminal.
7. The next component to implement will allow the player to provide names for the five party members. First use the input function to get the name for the party leader, storing it in a variable, and then print this back to the terminal before asking for (and storing) the remaining names. Please remember that you are not permitted to use repeating control structures (i.e., loops) or collections (e.g., lists, dictionaries, etc.) for this assignment, so every party member name must be stored in its own variable.
8. The last component will be an arithmetic "pipeline" allowing for the purchase of initial supplies. For the moment, assume that the initial amount of currency granted to your player is your student number, and store that value in a variable. When you completed the "Rough Work for Assignment 1" activity you registered the names of the different types of resources your player would manage, so for each of those you must print information to the terminal (about what the supply is and how much each unit costs), get input for how many units the player would like to purchase (storing this value in a variable after changing it to an integer data type), and then deduct the product of the cost per unit and the number of units requested from the total amount of currency that player has. Make sure you are using f-strings to print the "running" total (i.e., the total "bill so far") to two decimal places after purchasing each type of resource.

Do not forget that you are required to include descriptive comments throughout your program , describing the different components that have been accomplished in your submission.