

# CS118 Programming Assignment

## Menus

For many programs, it is necessary to provide a number of options to the user which describe the capabilities of the software. As programs become more mature, there can be many options available.

In a simple program, it is quite easy to provide the necessary prompts as execution of the program progresses. But as we develop more complex programs, it helps to begin to think of the program as “blocks” of code. Depending on the needs of the user, some of the blocks may get executed, while many will not.

A “menu” is a method for allowing the user to choose which part(s) of the program s/he wishes to execute. Programs can have a single (“main”) menu, or they can have a series of “cascading” menus – choosing an option on one menu can lead to more options on a separate menu. Menu options typically offer either single letters or numbers to indicate the choice of the user. For convenience, we will use numbers as choices from which our user can select:

```
Please choose from this menu:
```

1. Do this
2. Do that
3. Do something else
4. Do something weird
5. Exit the program

```
Your choice (1-5)?
```

In this assignment, you are to develop a couple of CLI (command line interface) menus – these are menus which are displayed in the shell [as opposed to using a GUI (graphical user interface) with dialog windows]. The main menu should consist of five options (your choice – have fun with it!). The first option of the main menu should lead to another menu with five new menu items (of your choice).

There are a few restrictions:

1. All screen output should be performed using `print()`. Use escape sequences as necessary to get good-looking output.
2. The main menu should have “Exit the program” as the last option. If the user chooses this option, the program should terminate. Termination should occur through running out of code, not the use of any special function.
3. The secondary menu should have “Return to main menu” as the last option. If the user chooses this option, the program should return to the main menu where the user can continue to choose.
4. Make each secondary menu item (except the last) print something to the screen and then return to the secondary menu. You can use the `raw_input()` function without collecting the return value will allow you to pause execution until the user hits ENTER.
5. On the main menu: Make sure the first main menu item leads to the secondary menu. You can have the others lead to menus if you wish, but the grader will only check the first menu item.
6. Do not attempt to “call the main program”. Your secondary menu should return automatically to the main menu.

Validate the user input. You may assume the user will only enter a number, but if the user chooses a number other than a valid menu item your program should “trap the user” and ask him/her for a different choice, repeating this until the user enters a valid option.

I recommend that you get the main menu working ENTIRELY before working on the secondary menu – you will find some duplication present that can be helpful.

## Example run:

Please choose:

1. Option such
2. Option such and such
3. Option such and such and such
4. Option 4\*such
5. Exit the program

Your choice? (1-5): 1

Please choose:

1. Option X
2. Option Y
3. Option Z
4. Option XYZ
5. Return to main menu

Your choice? (1-5): 1

You choose option 1!

Press ENTER to continue...

Please choose:

1. Option X
2. Option Y
3. Option Z
4. Option XYZ
5. Return to main menu

Your choice? (1-5): 5

Please choose:

1. Option such
2. Option such and such
3. Option such and such and such
4. Option 4\*such
5. Exit the program

Your choice? (1-5): 5

>>