CS118
Menu 1

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 command window menu. This main menu should consist of five options (your choice – have fun with it!).

There are a few restrictions:

1. All screen output should be performed using `print()`. Use one `print()` for each line of printed output. Use escape sequences as necessary to get good-looking output.

2. All menu options should begin with a number and your program should process the input as a number, not a string.

3. For each menu option chosen have the program print a nice message for the user.

4. The main menu should have "Exit the program" as the last option. If the user chooses this option, the program should terminate.

5. For those familiar, please do not "loop" the menu. We will be modifying the program later.

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 provide an error message and gracefully exit the menu.

Use an `IF/ELIF/ELSE` statement for processing the menu.

Example runs:

```
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
You chose option 1!



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)? 6
Invalid option!



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)? 5
Now exiting the program...
```