Programming Assignment 4

Objectives:

- Practice reading files and functions
- **DON'T USE ANYTHING WE HAVEN'T COVERED IN CLASS!**

Assignment:

In this assignment, you are going to write a program that gives the user name suggestions for a Dungeons & Dragons character based on the character's race and class. The names are stored in a file called characters.txt. Each line of the file is in the following format:

- Name,Race,Class

You will display the different races and classes the user can choose for their character (see sample execution for the list of races and classes). The user will enter 1 – 9 for their race and class choice. Then, using the user's choices, read the file, and display the names that have the same race and class as the user entered. If none of the names have the same race in class, display that no names were found.

Functions:

You must write the following functions in your program. They must match the specifications given below.

1. **main**
   a. No parameters
   b. Calls other functions. Gets user input.
   c. No return
2. **displayInstructions**
   a. No parameters
   b. Displays instructions for user at beginning of program.
   c. No return
3. **displayRaceOptions**
   a. No parameters
   b. Displays race options user can choose.
   c. No return
4. **displayClassOptions**
   a. No parameters
   b. Displays class options user can choose.
   c. No return
5. **determineRace**
   a. 1 parameter: user's choice of race
   b. Determines the name of the race based on the 1 – 9 the user entered.
   c. Returns race name.
6. **determineClass**
   a. 1 parameter: user's choice of class
   b. Determines the name of the class based on the 1 – 9 the user entered.
   c. Returns class name.
7. **displayCharacters**
   a. 2 parameters: race name, class name
   b. Displays all names associated with the given race and class.
   c. No return

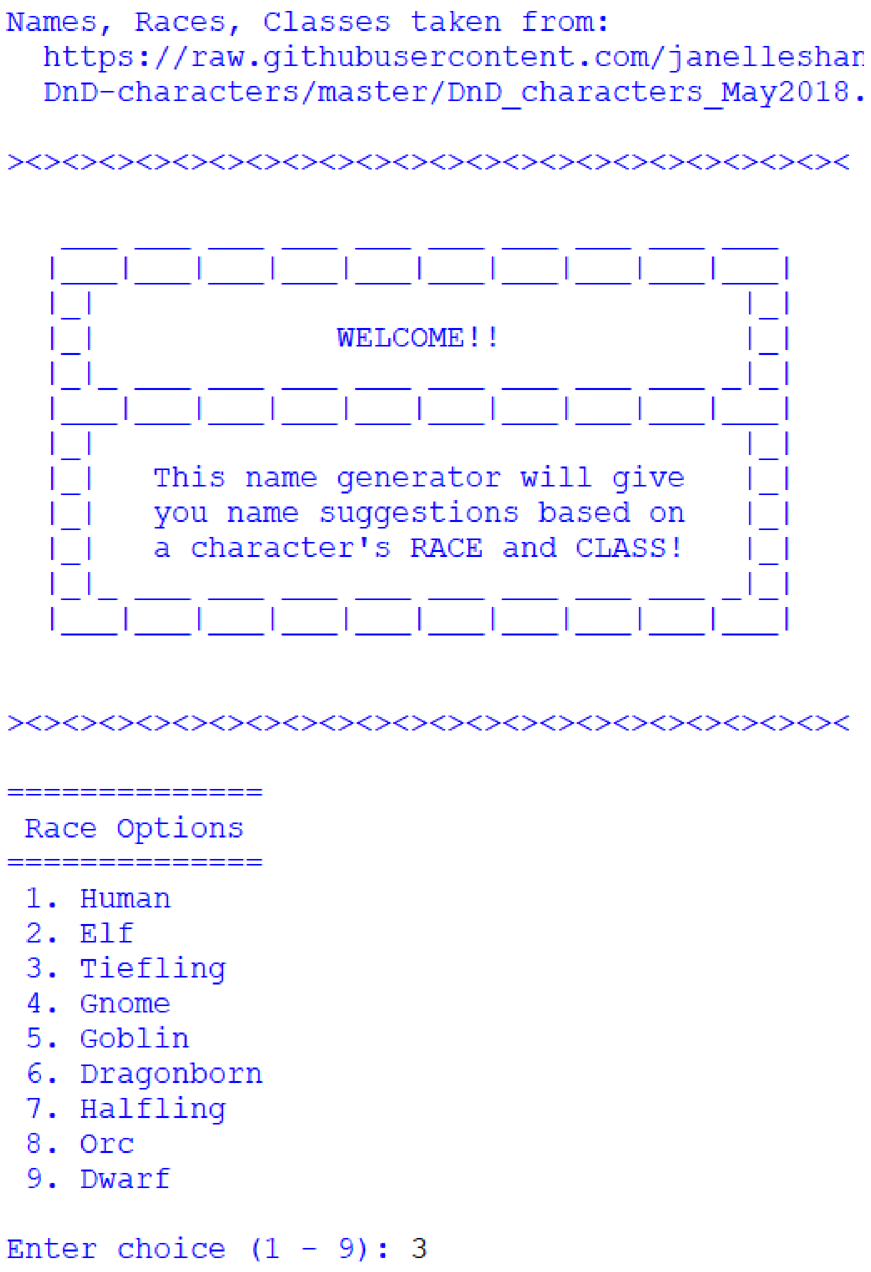
Note: Names, races and classes, taken from:
https://raw.githubusercontent.com/janelleshane/DnD-characters/master/DnD_characters_May2018.txt

Test Cases

As you write your program, you will need to test it, especially your decision structures. Download the Test Cases file from Moodle and fill in each row with the results of the tests in the file. It doesn't matter if the test is successful or not; the point is to document the testing process.

Sample Executions:

Sample Execution 1:



```
Names, Races, Classes taken from:
  https://raw.githubusercontent.com/janelleshan
  DnD-characters/master/DnD_characters_May2018.

><><><><><><><><><><><><><><><><><><><><><><><

   ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
  |___|___|___|___|___|___|___|___|___|___|
  |_|                                   |_|
  |_|             WELCOME!!             |_|
  |_|_ ___ ___ ___ ___ ___ ___ ___ ___ _|_|
  |___|___|___|___|___|___|___|___|___|___|
  |_|                                   |_|
  |_|   This name generator will give   |_|
  |_|   you name suggestions based on   |_|
  |_|   a character's RACE and CLASS!   |_|
  |_|_ ___ ___ ___ ___ ___ ___ ___ ___ _|_|
  |___|___|___|___|___|___|___|___|___|___|

><><><><><><><><><><><><><><><><><><><><><><><

==============
 Race Options
==============
 1. Human
 2. Elf
 3. Tiefling
 4. Gnome
 5. Goblin
 6. Dragonborn
 7. Halfling
 8. Orc
 9. Dwarf

Enter choice (1 - 9): 3
```

```
===============
 Class Options
===============
 1. Cleric
 2. Druid
 3. Wizard
 4. Paladin
 5. Bard
 6. Rogue
 7. Ranger
 8. Fighter
 9. Barbarian

Enter choice (1 - 9): 8

><><><><><><><><><><><><><><><><><><><><><><><

========================
 Tiefling/Fighter Names
========================

 - No Names Found.

><><><><><><><><><><><><><><><><><><><><><><><
```

Sample Execution 2:

```
Names, Races, Classes taken from:
  https://raw.githubusercontent.com/janelleshane/
  DnD-characters/master/DnD_characters_May2018.txt

><><><><><><><><><><><><><><><><><><><><><><><


   ____ ____ ____ ____ ____ ____ ____ ____ ____ ____
  |____|____|____|____|____|____|____|____|____|____|
  |_|                                             |_|
  |_|            WELCOME!!                        |_|
  |_|_  ____ ____ ____ ____ ____ ____ ____ ____  _|_|
  |____|____|____|____|____|____|____|____|____|____|
  |_|                                             |_|
  |_|   This name generator will give             |_|
  |_|   you name suggestions based on             |_|
  |_|   a character's RACE and CLASS!             |_|
  |_|_  ____ ____ ____ ____ ____ ____ ____ ____  _|_|
  |____|____|____|____|____|____|____|____|____|____|


><><><><><><><><><><><><><><><><><><><><><><><
```

```
==============
 Race Options
==============
 1. Human
 2. Elf
 3. Tiefling
 4. Gnome
 5. Goblin
 6. Dragonborn
 7. Halfling
 8. Orc
 9. Dwarf

Enter choice (1 - 9): 6


===============
 Class Options
===============
 1. Cleric
 2. Druid
 3. Wizard
 4. Paladin
 5. Bard
 6. Rogue
 7. Ranger
 8. Fighter
 9. Barbarian

Enter choice (1 - 9): 6

><><><><><><><><><><><><><><><><><><><><><><><

========================
 Dragonborn/Rogue Names
========================
 - Bihu
 - Zandri Sonsalor
 - Cecil
 - Chance

><><><><><><><><><><><><><><><><><><><><><><><
```

Requirements:

- Use an updated comment block
- Your program should use the following comment block at the very beginning of your program.

```
# Name:  Your name                    Date Assigned: Fill in
#
# Course:  2000-Sec #                 Date Due:  Fill in
#
# File name:  Fill in
#
# Program Description:  Brief description of what the program does.
```

- Use appropriate comments throughout the program
- Make good use of whitespace
- Output must be neat, logical, and easy to understand.

Deliverables:

- Upload the file that has your source code to Moodle with the extension .py. The file should be from **script** mode, NOT interactive mode. Upload the Excel file with your Test Cases.

Grading:

Total Points	**10 points**
main function is correct	1.5 points
displayInstructions is correct	0.5 point
displayRaceOptions is correct	1 point
displayClassOptions is correct	1 point
determineRace is correct	1.5 points
determineClass is correct	1.5 points
displayCharacters is correct	2 points
Test Cases	**1 point**
Not enough comments / whitespace.	**-1 point**
Output is difficult to read / understand or messy.	**-1 point**
Doesn't have good variable / function names.	**-1 point**