Assignment 5 - The Quiz

38 pts

In this assignment, you are creating a single program that has been broken into smaller components. The program will be able to create multiple choice and true/false questions used to build a simple quiz. The questions will be stored in a text file, and you will be able to take the quiz and determine your grade.

In [73]:
```
# display your student name and id with some unicode characters
# be imaginative!
print("Name: Yong ₩U0001F600eung R₩U0001F497o, StudentID: w0₩u00004₩uFE0F₩u20E34₩u00007₩uFE0F₩u20E34₩u00004₩uFE0F₩u20E32
```

```
Name: Yong 😀eung R💗o, StudentID: w04️⃣47️⃣44️⃣2
```

Name: Jane D⛬e, StudentID: w1️⃣2️⃣3️⃣4️⃣5️⃣6️⃣

Part A - Run the Quiz (15pts)

For this part of the assignment, you will create a sample quiz file and then run the quiz to display the questions.

The quiz file has this structure:

<question type>,<question>,<answer>,<additional question details>

True/False questions look like this:

```
TF,This is a sample question.,T
```

Multiple choice questions look like this:

```
MC,This is a sample question.,2,Choice 1,Choice 2,Choice 3,Choice 4
```

To read a quiz file, you use the following command line parameters:

```
python quiz.py run <filename>
```

Example,

```
python quiz.py run quiz.txt
```

In [1]:

```
# start the program by creating a "main" function that reads in the two command
# line parameters the program should exit if there are invalid parameters.
# the first parameter will be either 'create' or 'run', the second parameter will
# be a filename. the function will call either "create_quiz" or "run_quiz"
# depending on the parameter passed
# 2 pts
import sys

# put your code here
def print_help_exit():
    print()
    print('Usage: quiz.py <create|run> <filename>')
    sys.exit(-1)

def main():
    if len(sys.argv) > 2:
        if sys.argv[1] == "create":
            create_quiz(sys.argv[2])
        elif sys.argv[1] == "run":
            run_quiz(sys.argv[2])
        else:
            print_help_exit()
    else:
        print_help_exit()
```

In [2]:

```
# make a function that create a sample quiz file that consists
# of 1 true/false question and 1 multiple choice question
# 2 pts

# put your code here
def create_sample_quiz(filename):
    questions = [
        ['TF', 'This is a sample question.', 'T'],
        ['MC', 'This is a sample question.', '2', 'Choice 1', 'Choice 2', 'Choice 3', 'Choice 4']
    ]

    f = open(filename, 'w')
    for r in questions:
        num = len(r) - 1
        for i in range(num):
            f.write(r[i])
            f.write(',')
        f.write(r[num])
        f.write('₩n')
    f.close()
```

In [5]:

```
# create a function that will take a question and an answer
# from a true/false question and display it to the user
# the function will take the user's response and return
# whether they answered it correctly as a boolean value
# 2 pts

# put your code here
def display_truefalse(question, answer):
    user_answer = input(question + "₩nIs this statement true or false? (T/F): ")
    if user_answer == answer:
        return True
    return False

sample_question = 'This is a sample question.'
sample_answer = 'T'
ret = display_truefalse(sample_question, sample_answer)
if ret:
    print("Correct")
else:
    print("Incorrect")
```

```
Correct
```

In [1]:

```
# create a function that will take a question, an answer, and choices
# and display a multiple choice question to the user
# the function will take the user's response and return whether they
# answered it correctly as a boolean value
# 3 pts

# put your code here
def display_multiplechoice(question, answer, choices):
    if len(choices) == 4:
        print(question)
        i = 1
        for select in choices:
            print("%i) %s" % (i, select))
            i = i + 1
        user_answer = input("Enter your selection: ")
        if user_answer == answer:
            return True
        return False
    else:
        raise ValueError

sample_question = 'This is a sample question.'
sample_answer = '2'
sample_choices = ['Choice 1', 'Choice 2', 'Choice 3', 'Choice 4']
ret = display_multiplechoice(sample_question, sample_answer, sample_choices)
if ret:
    print("Correct")
else:
    print("Incorrect")
```

```
This is a sample question.
1) Choice 1
2) Choice 2
3) Choice 3
4) Choice 4
Correct
```

In [8]:

```
# create a function that will take a line of quiz data as a parameter
# and call one of the previous display functions based on the 'type'
# of question (defined by the first value in the line of quiz data)
# the function should return the boolean values returned from the
# display functions
# 4 pts

# put your code here
def display_question(line):
    data = line.rstrip('₩n').split(',')
    if data[0] == 'TF':
        ret = display_truefalse(data[1], data[2])
    else:
        ret = display_multiplechoice(data[1], data[2], *[data[3:7]])
    return ret
```

In [9]:

```
# create a function that will open the file for reading
# the function will read in each line of the file
# and call "display_question" with the contents of a line
# the function should return the total number of correct answers
# 2 pts

# put your code here
def run_quiz(filename):
    correct = 0
    f = open(filename)
    for line in f:
        ret = display_question(line)
        if ret:
            correct = correct + 1
    f.close()
    return correct
```

Part B - Create the Quiz (13pts)

For this part of the assignment, you will create questions for the quiz and generate a new quiz file.

To create a quiz file, you use the following command line parameters:

```
python quiz.py create <filename>
```

Example,

```
python quiz.py create quiz.txt
```

In [8]:

```
# make a function that creates a line of the quiz file
# for a true/false question. the function will ask the
# user to enter the question and what the correct answer is
# 2 pts

# put your code here
def create_truefalse():
    question = input("Enter the question: ")
    answer = input("Enter the answer (T or F): ")
    return f"TF,{question},{answer.upper()}"
```

In [9]:

```
# make a function that creates a line of the quiz file
# for a multiplel choice question. the function will ask
# the user to enter the question, the different choices
# to pick from (ENTER to end) and the correct answer
# 4 pts

# put your code here
def create_multiplechoice():
    choices = list()
    question = input("Enter the question: ")
    while True:
        possible_answer = input("Enter a possible answer (Enter to end): ")
        if possible_answer == '':
            break
        choices.append(possible_answer)
    str_choices = ",".join(choices)
    answer = input("Which one is the correct answer: ")
    return f"MC,{question},{answer},{str_choices}"
```

In [10]:

```
# make a function that will ask the user which type of question
# they want to create (MC or TF), then call the appropriate
# create function above. the function will return the line
# of data that is returned from the other create functions
# 3 pts

# put your code here
def create_question():
    input_type = input("What type of question do you want create (MC, TF or Enter to end)? ")
    type = input_type.upper()
    if type == 'TF':
        question = create_truefalse()
    elif type == 'MC':
        question = create_multiplechoice()
    elif type == '':
        question = type
    else:
        question = 'invalid'
    return question
```

In [10]:

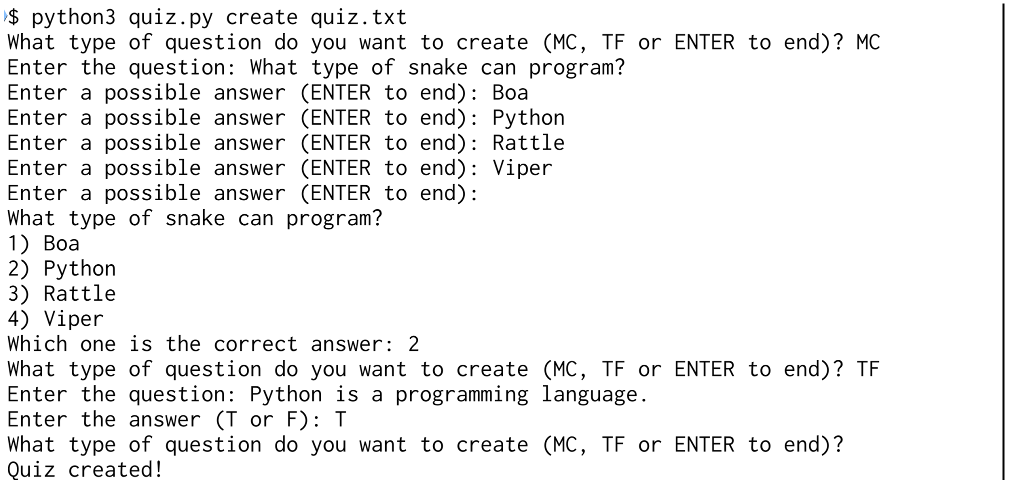
```
# make a function that creates a new, empty file using the filename passed in as
# a parameter. the function will then call 'create_question' to add new questions
# to the file until the user has finished
# (ensure that the file is closed properly)
# 4 pts

# put your code here
def create_quiz(filename):
    f = open(filename, 'w')
    question = create_question()
    if question != '':
        f.write(question)
        f.write('₩n')
    f.close()
```

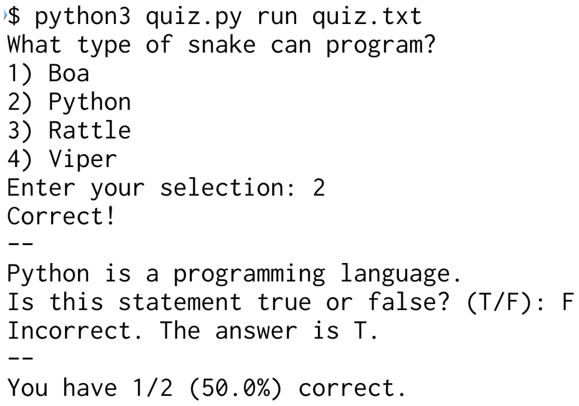
Part C - Pull Everything Together! (10pts)

For this part of the assignment, you will create a complete Python program using all the functions you have created. The final program will allow the user to either create or run a quiz, based on the command line arguments passed.

Submit the completed Python ".py" file, along with a PDF of the source file, and include a Word document showing screenshots of your program working (similar to the images below).



```
$ python3 quiz.py create quiz.txt
What type of question do you want to create (MC, TF or ENTER to end)? MC
Enter the question: What type of snake can program?
Enter a possible answer (ENTER to end): Boa
Enter a possible answer (ENTER to end): Python
Enter a possible answer (ENTER to end): Rattle
Enter a possible answer (ENTER to end): Viper
Enter a possible answer (ENTER to end):
What type of snake can program?
1) Boa
2) Python
3) Rattle
4) Viper
Which one is the correct answer: 2
What type of question do you want to create (MC, TF or ENTER to end)? TF
Enter the question: Python is a programming language.
Enter the answer (T or F): T
What type of question do you want to create (MC, TF or ENTER to end)?
Quiz created!
```



```
$ python3 quiz.py run quiz.txt
What type of snake can program?
1) Boa
2) Python
3) Rattle
4) Viper
Enter your selection: 2
Correct!
--
Python is a programming language.
Is this statement true or false? (T/F): F
Incorrect. The answer is T.
--
You have 1/2 (50.0%) correct.
```

In [11]: