CIS 125 Principles of Programming Logic
Exam #3: Repetition (Loop) Statements ANSWER KEY

Directions

Complete the following three Python programs worth a total of 100 points. Program #1 is worth 60 pts. Program #2 is worth 25 pts. Program #3 is worth 15 pts.

Submit (upload) the .py for each into MoodleRooms under Exam #3. They must be worked on individually and are due by the end of class. Late submissions cannot be accepted. Partial credit may be provided. The exam is open book, open note, open Internet. However, you cannot receive live assistance from anyone or post questions on a discussion forums, i.e. any resources you must already exist.

Academic Honestly Policy

Students are expected to uphold the school's standard of conduct relating to academic honesty. It is imperative that standards of academic integrity be upheld for the best interest of the student, college, community, and industry. Therefore, any instances of academic dishonesty/cheating (receiving or giving assistance to classmates or any live person) will result in an immediate submission of a failing course grade to the college records system. Violations resulting in a failing grade for the course will be forwarded to the Office of the Registrar.

Program #1: (60 pts)

Create a Python program named **exam3-1.py**.

- The program will ask the user for two user inputs in the **main part** of the program – a max value and a step/increment value for a for loop
- Pass these two values to a user-defined function. Name this function **`numLoop`**
- Inside the loop, create a for loop that outputs the numbers from 0 to the max value in increments of step value. See sample run below.

Sample run (user input of 100 for max value and 10 for step value):

```
Enter max value: 100
Enter step value: 10
Number:  0
Number:  10
Number:  20
Number:  30
Number:  40
Number:  50
Number:  60
Number:  70
Number:  80
Number:  90
Number:  100
```

```
def numNum(max,step):
    for x in range(0, max+1, step):
        print("Number: ", x)

max = int(input("Enter max value: "))
step = int(input("Enter step value: "))
numNum(max,step)
```

Program #2: (25 pts)

Create a Python program named **exam3-2.py** that does the following:

- A user-defined function is not needed for this program.
- Create a while loop
- Inside the while loop, prompt the user to enter a number of **meters they want to convert to inches**, or a value of zero or negative number of they want to exit the program/loop (this is a sentinel value)
- If the user does not enter a zero or negative value, convert the entered numeric value of meters to inches and display the output
- The loop will then run again and re-prompt the user (until the user enters 0 or a negative value to end loop).
- See the same run below and ensure your programs runs the same

Sample run (1 and 0 were entered as user input):

Enter meters to convert to inches (enter zero or negative number to stop): 1
1.0 meters is equal to 39.37 inches.
Enter meters to convert to inches (enter zero or negative number to stop): 0

Method #1

```
while True:
    meters = float(input("Enter meters to convert to inches (enter zero or negative number
to stop): "))
    if meters > 0:
        inches = meters * 39.37
        print (meters, "meters is equal to", inches, "inches.")
    else:
        break
```

Method #2

```
meters = 1
while meters > 0:
    meters = float(input("Enter meters to convert to inches (enter zero or negative number
to stop): "))
    if meters > 0:
        inches = meters * 39.37
        print (meters, "meters is equal to", inches, "inches.")
```

Method #3

```
meters = 1
while meters > 0:
    meters = float(input("Enter meters to convert to inches (enter zero or negative number
to stop): "))
    if meters <= 0:
        break
    inches = meters * 39.37
    print(meters, "meters is equal to %.1f" % inches, "inches.")
```

Program #3: (15 pts)

Create a Python program named **exam3-3.py** that does the following:

- Uses a loop to display the menu below and requires valid user input (1-3).
- If invalid user input is provided of any data type, the message shown below will display and the loop and user input will be run again.
- If menu options 1 or 2 are chosen, the appropriate stub message will be displayed and the program will pause until the user presses [Enter] after which the main menu will display again. A pause until the [Enter] key is pressed can be accomplished with this:

  ```
  input("\nReview policy stub\n")
  ```

- If menu option 3 is chosen, the message "Goodbye" will display and the program and loop will end.
- See sample run below and create a program that will run in this manner with the provided user input.

Sample run:

```
   MAIN MENU
----------------------
1. Review your policy
2. File a claim
3. Exit

Please enter menu choice (1-3): 1

Review policy stub

   MAIN MENU
----------------------
1. Review your policy
2. File a claim
3. Exit
```

Please enter menu choice (1-3): 2

File claim stub

MAIN MENU
---
1. Review your policy
2. File a claim
3. Exit

Please enter menu choice (1-3): dd

Invalid menu choice. Please re-enter.

MAIN MENU
---
1. Review your policy
2. File a claim
3. Exit

Please enter menu choice (1-3): 3

Goodbye

```
while True:
    print("     MAIN MENU")
    print("----------------------")
    print("1. Review your policy")
    print("2. File a claim")
    print("3. Exit")
    option = input("\nPlease enter menu choice (1-3): ")
    if option == "1":
        input("\nReview policy stub\n")
    elif option == "2":
        input("\nFile claim stub\n")
    elif option == "3":
        print("\nGoodbye\n")
        break
    else:
        print("\nInvalid menu choice.  Please re-enter.\n")
```