DATA STRUCTURES AND ALGORITHMS IN PYTHON

Assignment #1

TOPICS: PYTHON FUNDAMENTALS, VARIABLES, AND DECISIONS

Indian Creek School

DUE : *5 September 2023*

For problems that ask you to write a program, place your code in a .py file and write a comment at the beginning of the file that includes your name, date, and description of the contents of the file. Suggested file names appear in bold at the beginning of problems that ask you to write a program.

EXAMPLE:

```
# Name: Doug McNally
# Date: 2023-06-10
# Description: Lucky number checker

num = int( input("Pick a number: ") )
if num == 42:
     print("You entered the right number!")
else:
     print("You lose!")
```

PROBLEMS THAT REQUIRE YOU TO WRITE A PYTHON FILE: 2 – 7, 13 – 15
PROBLEMS THAT REQUIRE ANSWERS IN A PDF DOCUMENT: 8 – 12

1. Read section 1.1 in *Data Structures and Algorithms in Python.*
2. **Printing.py** Write a program using `print` that, when run, prints out a tic-tac-toe board. The purpose of this exercise is to make sure you understand how to write programs using your computing environment.

   Expected output:

   ```
      |  |
   --------
      |  |
   --------
      |  |
   ```

3. **Greet.py** Write a program that asks the user to enter their name and then greets them. For example if a user enters "Mr. McNally" your program should print out "Hello Mr. McNally!"
4. **Variables.py** Recall that variables are containers for storing information. For example,

   ```
   a = "Hello, world!"
   a = "and goodbye..."
   print(a)
   ```

   Output: `and goodbye...`
   The = sign is an assignment operator which tells the interpreter to assign the value "Hello, world!" to the variable `a`. The value of a after executing the first line above is "Hello, world!". But, after executing the second line, the value of `a` changes to "and goodbye...". Since we ask the program to print out `a` only after the second assignment statement, that is the value that gets printed. If you wanted to save the values of both strings, you should change the second variable to another valid variable name, such as `b`.
   Variables are useful because they can cut down on the amount of code you have to write. Write a program that prints out the tic-tac-toe board from exercise 2, but which uses variables to cut down on the amount of typing you have to do. *Hint* how many different variables should you need?
5. **LuckyNumber.py** Write a program that asks the user to enter a number. If the number they enter is your favorite number, tell them they picked the right number, otherwise tell them they lose.
   Hint: Remember that the `input()` command will always return a String value.
6. **NumberSign.py** Write a Python program that will take in a number from the user and print if it is positive, negative, or zero. Use a proper if/elif/else chain, don't just use three `if` statements.
7. **AssignGrade.py** Write a Python program that asks the user for a numerical grade in the range 0 to 100 and prints out the corresponding letter grade.

   A: 90 – 100, B: 80 – 89, C: 70 – 79, D: 60 – 69, F: < 60
8. Explain the difference between the = symbol in Python and mathematics.

9. Look at the code below. Write your prediction for what each line will print. If you are unsure about any of your answers, run the code in the Python shell to check if you are correct.

```
x = 5
y = 10
z = 10
print(x < y)
print(y < z)
print(x == 5)
print(not x == 5)
print(x != 5)
print(not x != 5)
print(x == "5")
print(5 == x + 0.00000000001)
print(x == 5 and y == 10)
print(x == 5 and y == 5)
print(x == 5 or y == 5)
print("3" == "3")
print(" 3" == "3")
print(3 < 4)
print(3 < 10)
print("3" < "4")
print("3" < "10")
print( (2 == 2) == "True" )
print( (2 == 2) == True )
print(3 < "3")
```

10. There are two things wrong with this code that tests if x is set to a positive value. One prevents it from running, and the other is subtle. Make sure the if statement works no matter what x is set to. Identify both issues.

```
x == 4
if x >= 0:
    print("x is positive.")
else:
    print("x is not positive.")
```

11. What three things are wrong with the following code?

```
x = input("Enter a number: ")
if x = 3
    print("You entered 3")
```

12. There are four things wrong with this code. Identify all four issues.

```
answer = input("What is the name of Dr. Bunsen Honeydew's assistant? ")
if a = "Beaker":
    print("Correct!")
    else
    print("Incorrect! It is Beaker.")
```

13. **SoundLevel.py** The decibel (dB) unit is used for describing how loud sounds are. Sound level in units of decibel is related to the sound pressure of the sound. The following table gives descriptions for certain sound levels.

Threshold of pain	130 dB
Possible hearing damage	120 dB
Jack hammer 1 meter away	100 dB
Traffic on a busy road 10 meters away	90 dB
Normal conversation	60 dB
Calm library	30 dB
Light leaf rustling	0 dB

Write a program that reads in a value in dB and then prints out the *closest* description from the list above.

14. **RopePhysics.py** A mass $m = 2$ kilograms is attached to the end of a rope of length $r = 3$ meters. The mass is whirled around at a high speed. The rope can withstand a maximum tension of $T = 60$ Newtons. Write a program that accepts a rotation speed v and determines whether such a speed will cause the rope to break.

The tension in the rope can be calculated using the following formula:

$$T = \frac{mv^2}{r}$$

Hint: Remember a number squared is just a number multiplied by itself, so if you have a variable `x`, you can compute x^2 with the following code: `x*x`.

Continued on the next page.

15. **QuizGame.py** At this point you know enough to write a simple trivia game. Your task is to create a quiz with at least 4 questions which will tell the user their score once they finish. Make sure to make fun and interesting questions so I can have fun taking your quiz.

    Now for requirements:

    1. At least 4 questions. You questions can be answered with
        - a number (e.g., What is 1+1?)
        - text (e.g., What is the capital of Maryland?)
        - a selection (e.g., Which of the following choices is correct? 1, 2, or 3?)
    2. Let the user know if they get each question correct. Print a message depending on their answer.
    3. Keep track of how many questions they get right and print their percentage score at the end.

    Some tips to help you:

    1. To create a running total of the number correct, create a variable to store this score. Set it to zero. With an if statement, add one to the variable each time the user gets a correct answer. (How do you know if they got it correct? Remember that if you are printing out "correct" then you have already done that part. Just add a line there to add one to the number correct.)
    2. Calculate the percentage by using a formula at the end of the game. Don't just add 25% for each question the user gets correct. If you add 25% each time, then you have to change the program 4 places if you add a 5th question. With a formula, you only need 1 change. You can calculate the percentage by taking

    $$\frac{\text{\# of questions right}}{\text{total \# of questions}} \times 100$$

    3. To print a blank line so that all the questions don't run into each other, use the following code: `print()`
    4. Remember the program can print multiple items on one line. This can be useful when printing the user's score at the end. `print("The value in x is", x)`
    5. Sometimes it makes sense to re-use variables. Rather than having a different variable to hold the user's answer for each question, you could reuse the same one.
    6. Use descriptive variable names. `x` is a terrible variable name. Instead use something like `number_correct`.

There is an example run of the program on the next page.

```
Quiz time!

How many books are there in the Harry Potter series? 7
Correct!

What is 3*2-1? 5
Correct!

What is the Indian Creek Mascot
1. Tiger
2. Eagle
3. Wildcat
4. Raven
? 2
Correct!

Who is on the front of a one dollar bill
1. George Washington
2. Abraham Lincoln
3. John Adams
4. Thomas Jefferson
? 2
No.

Congratulations, you got 3 answers right.
That is a score of 75.0 percent.
```