Lab #2 Grading Rubric and Instructions

This is the second lab for the 'Introduction to Computer Programming' course.

Please see the syllabus for information about when the work in this lab is due.

Lab Objective

The purpose of this lab is two-fold. Firstly, it is to give you a bit of enjoyable practice with the 'turtle' library in Python. Secondly, this lab will give you the opportunity to read library documentation and do a bit of research/ citation work for a variety of scripts. Having the ability to search for information in both official online documentation and elsewhere, and apply this information to your code, is crucial for programmers who want to be able to start 'answering their own questions' in regard to how to accomplish their tasks.

Instructions/ Deliverables

NOTE: These tasks can be completed in any order you like. See the **Grading Items** section below for the point distribution.

CITATION: Many of the exercises found here could possibly be seen as adaptations of exercises found in the online textbook "How to Think Like a Computer Scientist: Interactive Edition" By Jeffrey Elkner, Peter Wentworth, Allen B. Downey, Chris Meyers, and Dario Mitchell.

- **Available**: https://runestone.academy/ns/books/published/thinkcspy/index.html?mode=browsing
- **Accessed**: 1-28-2023
- The abbreviation 'thinkcspy' and the chapter/ section number will be used to indicate where similar exercises can be found. This citation will be placed next to the exercise title.
    - ex: [thinkcspy 2.13] indicates a similar exercise can be found in chapter 2, section 13.

CITATION: Some of the exercises found here are completely original to the instructor.

- The abbreviation 'MH' will be used to indicate these exercises. This citation will be placed next to the exercise title.
    - ex: [MH]

Lecture Notes/ Supplemental Readings:

- 'Check off' your notes in your Engineering Notebook for the following material with the TA/ Instructor. These notes should already be done before the start of the lab period.
    - Runestone chapters 4 and 5
        - **NOTE**: You do not need to complete any of the exercises at the end of the Runestone chapters. However, it would be helpful to you in the long term if you were to do so.
        - **NOTE**: You only need to take notes on the Runestone readings if you did *not* take lecture notes in class. The contents of these chapters are what was covered in the lectures.

`initials.py`: [MH]

- Explore the following documentation:
    - https://docs.python.org/3/library/turtle.html#turtle.color
- Use any of the turtle commands you like from the documentation above to draw a picture of the initials from your first and last name.
    - This will require drawing exactly two letters - the first letter of your first name, and the first letter of your last name.
    - Each letter of your initials should have an 'outline' that has a distinct color of your choosing.
    - The 'outline' of your initials should be 'filled in' with different colors of your choosing, such that the inner portions are clearly different from the 'outlines.'
    - The background of the drawing should be a different color than anything used in your letters.
    - The two letters should be separate from one another. Meaning you will have to 'pick up' the pen/ turtle after drawing the first letter, move it to a new location, and start drawing the second letter.
    - Use your imagination for how to accomplish all of this.
- Save your code, including your name, code creation date, lab number, and a brief description of what your code does, to a file called `initials.py`.

NOTE: All of the programming tasks below should follow a format similar to that seen in the `Example Script` section below. You will place your citations above your code with as much of the requested information as you can find. The only script that does not require a citation is your `initials.py` script.

`areaOfRectangle.py`: [thinkcspy 2.13]

- Prompt the user to enter two numerical values: the length of the base and the height of a rectangle.
- Use the values entered above to calculate the area of the triangle using the formula:

```
area = base * height
```

- You will need to perform an internet search and cite the definition of this formula in a comment. Include the URL, author (if available), date of the article, and the date you accessed the webpage.
- Print the results of your calculation to the terminal.
- Please see the 'Example Script' and 'Example Output' sections below for an example of what your code should look like.
- Save your code, including your name, code creation date, lab number, and a brief description of what your code does, to a file called `areaOfRectangle.py`.

`rectanglePerimeter.py`: [thinkcspy 2.13]

- Prompt the user to enter two numerical values: the length and width of a rectangle.
- Use the values entered above to calculate the perimeter of the rectangle using the formula:

```
perimeter = 2 * (length + width)
```

- You will need to perform an internet search and cite the definition of this formula in a comment. Include the URL, author (if available), date of the article, and the date you accessed the webpage.
- Print the results of your calculation to the terminal.
- Please see the 'Example Script' and 'Example Output' sections below for an example of what your code should look like.
- Save your code, including your name, code creation date, lab number, and a brief description of

what your code does, to a file called `rectanglePerimeter.py`.

`areaOfCircle.py`: [thinkcspy 2.13]

- Prompt the user to enter a numerical value: the radius of a circle.
- Use the value entered above to calculate the area of the circle using the formula:

```
area = math.pi * (radius ** 2)
```

- You will need to perform an internet search and cite the definition of this formula in a comment. Include the URL, author (if available), date of the article, and the date you accessed the webpage.
- **HINT:** You will need to import the `math` module at the top of your script, and use the `math.pi` value in your calculation.
- Print the results of your calculation to the terminal.
- Please see the 'Example Script' and 'Example Output' sections below for an example of what your code should look like.
- Save your code, including your name, code creation date, lab number, and a brief description of what your code does, to a file called `areaOfCircle.py`.

`circleCircumference.py`: [thinkcspy 2.13]

- Prompt the user to enter a numerical value: the radius of a circle.
- Use the value entered above to calculate the circumference of the circle using the formula:

```
circumference = 2 * math.pi * radius
```

- You will need to perform an internet search and cite the definition of this formula in a comment. Include the URL, author (if available), date of the article, and the date you accessed the webpage.
- Print the results of your calculation to the terminal.
- Please see the 'Example Script' and 'Example Output' sections below for an example of what your code should look like.
- Save your code, including your name, code creation date, lab number, and a brief description of what your code does, to a file called `circleCircumference.py`.

`distanceSpeedTime.py`: [thinkcspy 2.13]

- Prompt the user to enter two numerical values: speed in meters per second and time in seconds.
- Use the values entered above to calculate the distance traveled using the formula:

```
distance = speed * time
```

- You will need to perform an internet search and cite the definition of this formula in a comment (it is sometimes called the 'distance formula'). Include the URL, author (if available), date of the article, and the date you accessed the webpage.
- Print the results of your calculation to the terminal.
- Please see the 'Example Script' and 'Example Output' sections below for an example of what your code should look like.
- Save your code, including your name, code creation date, lab number, and a brief description of what your code does, to a file called `distanceSpeedTime.py`.

`velocityAccelerationTime.py`: [thinkcspy 2.13]

- Prompt the user to enter two numerical values: initial velocity in meters per second and acceleration in meters per second squared.
- Also, prompt the user to enter the time in seconds.
- Use the values entered above to calculate the final velocity using the formula:

```
final_velocity = initial_velocity + (acceleration * time)
```

- You will need to perform an internet search and cite the definition of this formula in a comment (it comes from Newtonian physics). Include the URL, author (if available), date of the article, and the date you accessed the webpage.
- Print the results of your calculation to the terminal.
- Please see the 'Example Script' and 'Example Output' sections below for an example of what your code should look like.
- Save your code, including your name, code creation date, lab number, and a brief description of what your code does, to a file called `velocityAccelerationTime.py`.

`calculateVoltage.py`: [thinkcspy 2.13]

- Prompt the user to enter two numerical values: current and resistance.
- Use the values entered above to calculate the average score using the formula:

```
voltage = current * resistance
```

- You will need to perform an internet search and cite the definition of this formula in a comment (it comes from Ohm's law). Include the URL, author (if available), date of the article, and the date you accessed the webpage.
- Print the results of your calculation to the terminal.
- Please see the 'Example Script' and 'Example Output' sections below for an example of what your code should look like.
- Save your code, including your name, code creation date, lab number, and a brief description of what your code does, to a file called `calculateVoltage.py`.

`calculateResistance.py`: [thinkcspy 2.13]

- Prompt the user to enter two numerical values: voltage and current.
- Use the values entered above to calculate power consumption using the formula:

```
resistance = voltage / current
```

- You will need to perform an internet search and cite the definition of this formula in a comment (it comes from Ohm's law). Include the URL, author (if available), date of the article, and the date you accessed the webpage.
- Print the results of your calculation to the terminal.
- Please see the 'Example Script' and 'Example Output' sections below for an example of what your code should look like.
- Save your code, including your name, code creation date, lab number, and a brief description of what your code does, to a file called `calculateResistance.py`.

`calculateCurrent.py`: [thinkcspy 2.13]

- Prompt the user to enter two numerical values: voltage and resistance.
- Use the values entered above to calculate the average score using the formula:

```
current = voltage / resistance
```

You will need to perform an internet search and cite the definition of this formula in a comment (it comes from Ohm's law). Include the URL, author (if available), date of the article, and the date you accessed the webpage.

- Print the results of your calculation to the terminal.
- Please see the 'Example Script' and 'Example Output' sections below for an example of what your code should look like.
- Save your code, including your name, code creation date, lab number, and a brief description of what your code does, to a file called `calculateCurrent.py`.

`annualPercentageRate.py`: [thinkcspy 2.13]

- Prompt the user to enter four numerical values: interest charges, fees, loan amount, and the number of days in the loan term.
- Use the values entered above to calculate the annual percentage ratee (APR) of a loan using the formula:

```
apr = (((interest_charges + fees) / loan_amount) / days_in_term) * 100
```

- You will need to perform an internet search and cite the definition of this formula in a comment. Include the URL, author (if available), date of the article, and the date you accessed the webpage.
- Print the results of your calculation to the terminal.
- Please see the 'Example Script' and 'Example Output' sections below for an example of what your code should look like.
- Save your code, including your name, code creation date, lab number, and a brief description of what your code does, to a file called `annualPercentageRate.py`.

`compoundAmount.py`: [thinkcspy 2.13]

- Prompt the user to enter three numerical values: principal amount, interest rate (as a percentage - excluding the 'percent sign'), the number of times the interest compounds in a year, and a time in years.
- Use the values entered above to calculate compound interest using the formula:

```
accrued_amount = principal * (1 + (rate/100) / number_compounds)**(number_compounds
* time)
```

- You will need to perform an internet search and cite the definition of this formula in a comment (it calculates the total principal + compound interest of a loan). Include the URL, author (if available), date of the article, and the date you accessed the webpage.
- Print the results of your calculation to the terminal.
- Please see the 'Example Script' and 'Example Output' sections below for an example of what your code should look like.

- Save your code, including your name, code creation date, lab number, and a brief description of what your code does, to a file called `compoundAmount.py`.

Attendance:

- If you have completed all of your tasks for the lab, you may work on any of the 'Additional Resources for Study' found in the Canvas announcement of the same name.
    - **NOTE**: If you leave early, you will not receive the 'attendance points' for the lab.

Optional Readings

NOTE: These readings are not required. However, they may provide a bit of interest/ insight into the broader world of Computer Science. Please complete the rest of your lab tasks before doing these readings. You do not need to take notes on these in your Engineering Notebook.

Units of Measure in Computing - by: Codecademy Team

- Available: https://www.codecademy.com/article/units-of-measure-in-computing

How Do Temperature Sensors Work? - by: AtlasScientific

- Available: https://atlas-scientific.com/blog/how-do-temperature-sensors-work/

10 Most Common Types of Cyber Attacks - by: Kurt Baker

- Available: https://www.crowdstrike.com/cybersecurity-101/cyberattacks/most-common-types-of-cyberattacks/

5 Used Car Scams and How to Avoid Them - by: Ben Luthi

- Available: https://www.experian.com/blogs/ask-experian/used-car-scams/

Why AI Is Bad For Gaming & The Video Game Industry - by: HUMAN Thomas Knutsen

- Available: https://aiconsequences.com/why-ai-is-bad-for-gaming/

Files Provided

None

Example Script

mySimpleExample.py

```
# Matthew Holman              1-28-2023
# Lab Week 3 - This example code calculates the sum of two numbers the user inputs.

# CITATION - URL: ?
# CITATION - Author: ?
# CITATION - Date Written/ Posted: ?
# CITATION - Date Accessed: ?

a = int(input("Enter value 'a': "))
b = int(input("Enter value 'b': "))
c = a + b
print(f"The result of {a} + {b} = {c}")
```

HINT: Study the example of the 'f-string' used in the code above. Do see how the values in the 'curly brackets' (`{`, `}`) relate to the variables `a`, `b`, and `c`? Do you notice the letter `f` immediately to the left of the quotation marks? Consider experimenting with this concept a bit on your own.

Example Output

Running: `python mySimpleExample.py`

```
Enter value 'a': 2
Enter value 'b': 3
The result of 2 + 3 = 5
```

Grading Items

- (**Lecture Notes/ Supplemental Readings**) Has the student taken notes on the listed material, and shown their notes in their Engineering Notebook to the TA/ Instructor?: ______ / 20
- (**initials.py**) Has the student completed the task above, and saved their work to a file called `initials.py`?: ______ / 20
- (**areaOfRectangle.py** and **rectanglePerimeter.py**) Has the student completed the task above, and saved their work to files called `areaOfRectangle.py` and `rectanglePerimeter.py`?: ______ / 10
- (**areaOfCircle.py** and **circleCircumference.py**) Has the student completed the task above, and saved their work to files called `areaOfCircle.py` and `circleCircumference.py`?: ______ / 10
- (**distanceSpeedTime.py** and **velocityAccelerationTime.py**) Has the student completed the task above, and saved their work to a file called `distanceSpeedTime.py` and `velocityAccelerationTime.py`?: ______ / 10
- (**calculateVoltage.py** and **calculateResistance.py** and **calculateCurrent.py**) Has the student completed the task above, and saved their work to files called `calculateVoltage.py`, `calculateResistance.py`, and `calculateCurrent.py`?: ______ / 10
- (**annualPercentageRate.py** and **compoundAmount.py**) Has the student completed the task above, and saved their work to files called `annualPercentageRate.py` and `compoundAmount.py`?: ______ / 10
- (**Attendance**) Did the student attend the full lab meeting in person?: ______ / 10

TOTAL ______ / 100