

The purpose of this lab is to gain experience with basic Python data types, variables, operators and basic I/O operations.

The completed lab must be submitted via Blackboard NLT (no later than) the beginning time of the lab session immediately subsequent to the lab section in which it is assigned. If you have questions, use the Blackboard discussion forums (and instructor/TA office hours) to obtain assistance.

### **Lab Requirements:**

- The source code for this lab must be submitted in a file named using the following format:
  - *Firstname + underscore + lastname + underscore + labSectionNumber + underscore + lab + PANumber dot py*
  - **Example:** Josephine\_Student\_20x\_PA1.py
- The source code file must contain a **file header** formatted as specified below. The header information must be complete and accurate.
- The source code file should use **self-documenting code** and additional comments (as required) to improve code readability.

### **Source File Header:**

```
#-----
# source_file_name.py
# Student Name: Joe Student
# Assignment: Lab #1
# Submission Date: 09/09/9999
#-----
# Honor Code Statement: I received no assistance on this assignment that
#                       violates the ethical guidelines as set forth by the
#                       instructor and the class syllabus.
#-----
# References: (list of web sites, texts, and any other resources used)
#-----
# Comments: (a note to the grader as to any problems or uncompleted aspects of
#           of the assignment)
#-----
# Pseudocode: Provided in spec
#-----
# NOTE: width of source code should be < 80 characters to facilitate printing
#-----
```

**Lab Scenario:**

For this lab, you are required to create a console-based application that calculates Simple and Compound Interests for a given set of inputs. You will request the following inputs from the user.

P = principal amount

R = rate of interest in % (per annum)

Y = Time duration in Number of year

M = Number of additional Months (valid value for this input is < 12)

Once you have the inputs, you will use the following formulae to calculate Simple and Compound Interests and print them. For compound interest, assume that the interest is compounded annually.

$$\text{SimpleInterest} = P * N * r$$

$$\text{CompoundInterest} = P(1 + r)^N - P$$

Before applying the formulae to the inputs, you need to perform appropriate conversions. Please note that the formulae replace M, Y and R with N and r. Where N is the time period in years obtained by combining Y (years) and M (months). Therefore, N will have a fractional part if M is non-zero. Similarly r is obtained by dividing R by 100.

$$N = Y + M/12$$

$$r = R/100$$

---

**Lab Procedure:**

Design Your Solution: Before you start coding, think about the inputs, outputs and processing, and design your solution. Although, the design for this problem is quite simple, it will help you understand how to approach programming problems. Put down your solution as a sequence of steps, in plain English, providing as much detail as required. This sequence of steps is called pseudocode. Once you have the pseudocode, it is usually a trivial task to convert it into code. You are required to include the pseudocode in your submission. Refer to grading rubric on the next page for grading guidelines.

Coding: Use the pseudocode to write a python solution for the given task. But, before that, familiarize yourself with the different operators in python by running simple statements in python console. Once you develop an understanding of the basic constructs that you will be using in your code, you will be more confident in using them in your program.

Use meaningful variable names and provide appropriate prompts for input and output statements. Look at example solutions in your lab slides. And don't forget to use comments where necessary.

Testing with different input: Finally, when you have the code ready, test it using different inputs. Verify

the result by hand calculating the expected results. Also try invalid inputs that break your program or give weird results e.g. negative values or giving no inputs. Don't worry if your solution breaks when you provide *invalid* inputs. You will learn how to fix it in later labs.

---

**Grading Rubric:**

	Excellent	Average	Needs Improving	Points
Submission Details	Both file name and file header meet stated spec.	Either file name is incorrect or file header is missing sections/details	Both file name and file header is missing or are incorrectly implemented.	___/2
Pseudo code	Pseudocode is present and provides sufficient detail for a developer to accurately implement the process.	Pseudocode is present, but provides insufficient detail for a developer to completely implement the process.	Pseudocode is missing or is so general that it provides little assistance in implementing the process.	___/3
Comments & Self Documenting Code	Comments clearly demonstrate which sections of code are related to specific steps in the pseudocode and variable names clearly reflect what they represent.	Comments generally demonstrate which sections of code are related to specific steps in the pseudocode and variable names generally reflect what they represent.	Comments are missing or provide little assistance to the understanding of the code and/or variable naming conventions provide no insight into the referenced data.	___/3
Formula Implementation	Formula is implemented correctly, including any required data transformations.	Formula is generally implemented correctly, but contains minor flaws or required data transformations are not implemented.	Formula is significantly flawed, and required data transformations are not implemented.	___/5
I/O	All specified input and output are implemented.	All specified standard input and output are generally implemented with minor flaws.	Specified standard input and output are not implemented.	___/2
Overall				___/15