# Assignment 2

### Input, Processing and Output

### Due: September 27, October 3, 2024

The learning objectives of the assignment are:

- Build problem solving skills.
- Design an algorithm and implement it in code. The code should include:
  - clear variable definitions, constants, and data types
  - reading and casting of user input
  - mathematical operations
  - output using `f-strings`
- - Debugging

The problem to be solved is to write a pension-calculator program (similar to this one) to estimate an worker's retirement income. It will prompt the user to enter several pieces of information, and will then compute and display their expected pension amount for three different retirement ages.

## Retirement Income Calculation

We will use a simplified version of the pension calculation used by the Local Authorities Pension Plan (LAPP), a pension provider used by many Albertan employers, including MRU. The calculation is:

```
pension_income = average_of_best_3_annual_incomes x rate x years_of_service_at_retirement
```

Where rate is set by the pension provider. Assume that rate is 1.4%

You will perform the calculation using the following pieces of information as inputs:

- The worker's current age in years;
- The worker's current years of service;
- The three largest annual incomes the worker expects during their career.

Given this information you must compute pension income for three different retirement ages: 55, 60, and 65.

## Part 1 Hand Calculations Due Sept 27

Before writing code any code you must perform some sample pension calculations by hand for two scenarios. For each compute by hand (using a calculator) the retirement incomes for the worker at ages 55, 60 and 65. Show all your work. Note that getting from the inputs to the result requires some intermediate calculations: it is important to show all calculations step-by-step. When writing and debugging your program, you'll compare its outputs to your manual calculations to ensure it is working properly.

### *Scenario 1*

- worker's current age = 40
- worker's current years of service = 10
- worker's largest 3 expected incomes are $65,000.00, $68,000.00 and $72,000.00

### *Scenario 2*

- worker's current age = 60
- worker's current years of service = 20
- worker's largest 3 expected incomes are $110,000.90, $142,000.00 and $118,178.50

### *Part 1 Submission*

You can submit your hand calculations as a picture or scan of a handwritten document (as long as it is clearly readable), or you can create a pdf using the tool of your choice (I will post an introduction to LaTeX). Submit the file to D2L by the duedate.

## Part 2 Coding Due Oct 3

Write a Python program that computes a worker's estimated retirement income. When the program starts, it must prompt the user to enter the following values (in this order):

- The worker's current age in years;
- The worker's current years of service;
- The three largest annual incomes the user expects during their career (entered one at a time).

Your program will compute the worker's annual pension income for three different retirement ages: 55, 60, and 65. Results to be displayed in a table, formatted

as dollar values with comma-thousand-separators.

After running the program, your terminal window should look like this (values entered by the user are shown in bold)

```
Enter current age in years: 25
Enter current years of service: 2
Enter the largest expected annual income: 50000.0
Enter the second-largest expected annual income: 52000.50
enter the third-largest expected annual income: 55000

retirement age      retirement income
55                  $23,445.33
60                  $27,108.67
65                  $30,772.00
```

**NOTE**: your program's output table must use the same formatting (you may need to google some f-string format codes), and the program must request inputs in the same order as shown above.

***Starter script/file:***

The starter script contains a file called `main.py`. Write your code in this file. The comments in the file will help you structure and arrange your code, by prompting you where to put constants, inputs, processing code, outputs, and functions (if used).

***Restrictions:***

You must use only Python techniques taught in class. If in doubt, ask your instructor. You may use functions in your program if you wish—they may simplify your program, but are not necessary.

***Part 2 Submission:***

Submit your `main.py` script.

## Grading

Your submission be graded as follows:

| Item | Points | Notes |
|---|---|---|
| Sample calculations | 10 | Completeness, correctness, and clarity |
| Code runs without errors | 5 | |
| Code style | 5 | Appropriate use of constants, good variable names, comments, readable code, etc. |
| Input prompts | 5 | The 5 pieces of information are obtained from the user in the order shown above |
| Calculations are correct | 18 | Including correct casting where necessary |
| Output formatted correctly | 7 | Should be exactly as shown |
| TOTAL | 50 | |