FACULTY OF COMPUTING & INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY

كلية الحاسبات وتقنية المعلومات
جامعة الملك عبدالعزيز

Lab 4 - Elementary Programming II

Chapter:	2. Elementary Programming
Time:	80 Minutes
Week:	**5**
Date:	29/09/2019 – 03/10/2019

Lab 4

Objectives

- To familiarize students how to solve practical problems programmatically.
- To practice on elementary programming by using Python built-in data types, variables, constants, operators, expressions, and input and output.

Current Lab Learning Outcomes (LLO)

By completion of the lab the students should be able to

- Use the *input* function.
- Use variables.
- Use constants.
- Use arithmetic operators.
- Distinguish between float division operator and integer division operator.
- Write simple calculations.

Lab Requirements

- PyCharm (IDE).

Practice Activities with Lab Instructor (20 minutes)

Problem 1

Programming Exercises (2.17)

Body mass index (BMI) is a measure of health based on weight. It can be calculated by taking your weight in kilograms and dividing it by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI.

$$\text{BMI} = \frac{\text{Weight (kg)}}{\text{Height (m)}^2}$$

Note that one pound is 0.45359237 kilograms and one inch is 0.0254 meters.

Here is a sample run:

```
Enter weight in pounds: 95.5 <enter>
Enter height in inches: 50 <enter>
BMI is 26.857257942215885
```

Solution

Phase 1: Problem-Solving Phase:

1- Ask the user to enter the weight in pounds (weight_in_pounds).
   - Use the *input* function to read inputs from the user.
   - Use the *eval* function to evaluate the string input to numbers.
   - `weight_in_pounds = eval(input("message…"))`
2- Ask the user to enter the height in inches (height_in_inches).
   - `height_in_inches = eval(input("message…"))`
3- Convert the weight in pounds (weight_in_pounds) to kilograms (weight_in_kilograms).
   - Note: one pound is 0.45359237 kilograms. Define this value as a constant (ONE_POUND_IN_KILOGRAMS) to easily use it later in calculations.
   - `ONE_POUND_IN_KILOGRAMS = 0.45359237`
   - Multiply weight_in_pounds by ONE_POUND_IN_KILOGRAMS to get the weight in kilograms. **Why?** To convert a smaller unit to a larger unit, divide it by the number of smaller units which are needed to make larger unit. To convert from a larger unit to a smaller one, multiply.
   - `weight_in_kilograms = weight_in_pounds * ONE_POUND_TO_KILOGRAMS`
4- Convert the height in inches (height_in_inches) to meters (height_in_meters).
   - Note: one inch is 0.0254 meters. Define this value as a constant (ONE_INCH_IN_METERS) to easily use it later in calculations.

- `ONE_INCH_IN_METERS = 0.0254`
- Multiply height_in_inches by ONE_INCH_IN_METERS to get the height in meters. **Why?** To convert a smaller unit to a larger unit, divide it by the number of smaller units which are needed to make larger unit. To convert from a larger unit to a smaller one, multiply.
- `height_in_meters = height_in_inches * ONE_INCH_IN_METERS`

5- Calculate the body mass index (bmi).
   - Note: $a^2 = a \times a$
   - `bmi = weight_in_kilograms / (height_in_meters * height_in_meters)`

6- Display the results (bmi).

Phase 2: Implementation Phase:

1. Create a new project and name it "Lab 4".
2. Create a new file and name it "activity_1.py".
3. Write the following code in the file:

activity_1.py

```
1  # Ask the user to enter the weight in pounds
2  weight_in_pounds = eval(input("Enter weight in pounds: "))
3
4  # Ask the user to enter the height in inches
5  height_in_inches = eval(input("Enter height in inches: "))
6
7  # Convert the weight in pounds to kilograms
8  ONE_POUND_IN_KILOGRAMS = 0.45359237
9  weight_in_kilograms = ONE_POUND_IN_KILOGRAMS * weight_in_pounds
10
11 # Convert the height in inches to meters
12 ONE_INCH_IN_METERS = 0.0254
13 height_in_meters = height_in_inches * ONE_INCH_IN_METERS
14
15 # Calculate the body mass index (BMI)
16 bmi = weight_in_kilograms / (height_in_meters * height_in_meters)
17
18 # Display the result
19 print("BMI is", bmi)
```

Problem 2

Programming Exercises (2.19)

Write a program that reads in an investment amount, the annual interest rate, and the number of years, and displays the future investment value using the following formula:

$$futureInvestmentValue = investmentAmount \times (1 + monthlyInterestRate)^{numberOfMonths}$$

For example, if you enter the amount 1000, an annual interest rate of 4.25%, and the number of years as 1, the future investment value is 1043.33. Here is a sample run:

```
Enter investment amount: 1000 <enter>
Enter annual interest rate: 4.25 <enter>
Enter number of years: 1 <enter>
Accumulated value is 1043.33
```

Solution

Phase 1: Problem-Solving Phase:

1- Ask the user to enter the investment amount (investmentAmount).
   - `investmentAmount = eval(input("message…"))`
2- Ask the user to enter the annual interest rate (annualInterestRate).
   - `annualInterestRate = eval(input("message…"))`
3- Ask the user to enter the number of years (numberOfYears).
   - `numberOfYears = eval(input("message…"))`
4- Calculate the monthly interest rate (monthlyInterestRate).
   - Note that the annual interest rate (annualInterestRate) is in percentage (%). If you want to use it in a calculation, you have to remove the percentage (%) first by dividing the value (rate) by 100.
   - 1 Year = 12 Months
   - `monthlyInterestRate = (annualInterestRate / 100) / 12`
5- Calculate the number of months (numberOfMonths).
   - 1 Year = 12 Months
   - `numberOfMonths = numberOfYears * 12`
6- Calculate the future investment value (futureValue).
   - `futureValue = investmentAmount * ((1 + monthlyInterestRate) ** numberOfMonths)`
   - You can use the line continuation symbol (\) to split the statement line into two lines.

```
futureValue = investmentAmount * \
              ((1 + monthlyInterestRate) ** numberOfMonths)
```

7- Display the result (futureValue).

Phase 2: Implementation Phase:

1. Open the project “Lab 4” if it was not opened or create it if it was not existing.
2. Create a new file and name it “activity_2.py”.
3. Write the following code in the file:

activity_2.py

```
1  # Enter the investment amount
2  investmentAmount = eval(
3      input("Enter the investment amount, for example 120000.95: "))
4  # Enter yearly interest rate
5  annualInterestRate = eval(
6      input("Enter annual interest rate, for example 8.25: "))
7  # Enter number of years
8  numberOfYears = eval(
9      input("Enter number of years as an integer, for example 5: "))
10
11 # Calculate monthly interest rate
12 monthlyInterestRate = (annualInterestRate / 100) / 12
13 # Calculate the number of months
14 numberOfMonths = numberOfYears * 12
15 # Calculate the future investment value
16 futureValue = investmentAmount * \
17               ((1 + monthlyInterestRate) ** numberOfMonths)
18
19 # Display the result
20 print("Future value is", int(futureValue * 100) / 100.0)
```

Individual Activities (60 minutes)

Problem 3

Programming Exercises (2.5)

Write a program that reads the subtotal and the gratuity rate and computes the gratuity and total. For example, if the user enters 10 for the subtotal and 15% for the gratuity rate, the program displays 1.5 as the gratuity and 11.5 as the total.

Here is a sample run of the program:

```
Enter the subtotal and a gratuity rate: 15.69, 15 <enter>
The gratuity is 2.35 and the total is 18.04
```

Problem 4

Programming Exercises (2.20)

If you know the balance and the annual percentage interest rate, you can compute the interest on the next monthly payment using the following formula:

$$interest = balance \times \frac{annualInterestRate}{1200}$$

Write a program that reads the balance and the annual percentage interest rate and displays the interest for the next month. Here is a sample run:

```
Enter balance and interest rate (e.g., 3 for 3%): 1000, 3.5 <enter>
The interest is 2.91667
```

Extra Exercises (Homework)

From the Textbook

- Programming Exercises:
  - 2.12
  - 2.13
  - 2.14
  - 2.21

From MyProgrammingLab (https://pearson.turingscraft.com)

- 2.6
  - 51045
  - 51047
  - 51880
- 2.8
  - 51032
  - 51034
  - 51046
  - 51249
- 2.10
  - 51094
  - 51095
  - 51189

Upload Your Solutions

Upload your solutions of the lab activities to Blackboard.