|  |  |  |
| --- | --- | --- |
| **Programming 1 (PRG1)**  Diploma in IT / DS / CSF / IM / CICTP  Year 1 (2023/24) Semester 1 | | Week **2** |
|  |
| **Activities :** | **Operators and Interactivity String Formatting and Errors**  **File IO** | |

**OBJECTIVES**

At the end of this exercise, students should be able to perform the following:

* Write Python code to swap two values
* Evaluate Boolean expression
* Write Python program that involve user interactivity, arithmetic operators and math function

**IMPORTANT**

* Create a folder, **Week02**, on your hard disk.
* Save this word document as **Week02-YourName.docx** in the **Week02** folder created above.
* For each question, type your answer into the box provided below the question.
* For the questions that require you to write Python program, create the Python program with the given file name in the **Week02** folder created above. Do add the description, your name and student ID as comments at the beginning of each program.
* At the end of the session, save this word document, compress all the files in your **Week02** folder (i.e. the Python program files and this word document) and submit the zip file in POLITEMall.

**PART 1**

Activity 1

Given two integers num1 and num2, write Python statement(s) to swap the values in num1 and num2?

|  |
| --- |
| num1, num2 = num2, num1 |

Activity 2

Suppose x = 4, y = 3, z = 2, evaluate the following expressions:

|  |  |
| --- | --- |
| **Boolean Expression** | **Result** |
| x < y + z | True |
| y == 2 \* x + 3 | False |
| z <= x - y | False |
| z > x % z | True |
| x != y | True |

Activity 3

Suppose x = 1, y = 7, z = 14, evaluate the following expressions:

|  |  |
| --- | --- |
| **Boolean Expression** | **Result** |
| x <= 1 and y == 3 | False |
| x <= 1 or y == 3 | True |
| not (x <= 1) | False |
| not x <= 1 or y == 3 and z > 14 | False |
| not (x <= 1 or y == 3 and z > 14) | False |
| not (x <= 1 or y == 3) and z > 14 | False |
| not ((x <= 1 or y == 3) and z > 14) | True |

Activity 4 (TimeConversion.py)

Write a program that accepts the time in seconds as input and converts it into hours, minutes and seconds.

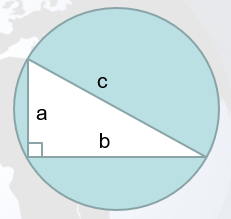
Sample output (value underlined is the user input):

|  |
| --- |
| Please enter the time to be converted, in sec: 8600  Time: 2 hr, 23 min 20 sec |

Hint: Use the integer division (//) and modulus (%) operators.

Activity 5 (Circle.py)

In the following diagram, *c* is the diameter of the circle. Given the lengths of *a* and *b*, write a program to calculate:



1. The length of *c*
2. The area of the circle

Sample output (values underlined are the user input):

|  |
| --- |
| Enter the length of a: 2.5  Enter the length of b: 4.6  The length of c is 5.235456045083369  The area of the circle is 21.52776365872405 |

**PART 2**

Activity 6 (DisplayTable.py)

Write a program that displays the following table:

Output:

|  |
| --- |
| Number Square Square root English  1 1 1.00 One  2 4 1.41 Two  3 9 1.73 Three  4 16 2.00 Four  5 25 2.24 Five |

Activity 7 (BMI\_BMR\_Calculator.py)

Open the file “Program Errors and Debugging.docx” and follow the steps given in the file to learn to use the IDLE Debugger.

After you have corrected the errors found in the program (BMI\_BMR\_Calculator.py) while debugging, trace the results again and complete the trace table below to record your inspection results of the variables when each statement is executed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Line Number** | **Variable**  **weight** | **Variable**  **height** | **Variable**  **bmi** | **Variable**  **age** | **Variable**  **bmr** |
| 5 | 55 | - | - | - | - |
| 6 | 55 | 1.7 | - | - | - |
| 8 | 55 | 1.7 | 32.35294117647059 | - | - |
| 13 | 55 | 1.7 | 32.35294117647059 | 40 | - |
| 15 | 55 | 1.7 | 32.35294117647059 | 40 | 365.625 |
|  |  |  |  |  |  |

Observe the output of the program. It does not produce correct result although the program runs smoothly. Identify and correct the *logic errors* in the program.

State the logic error and your correction below:

|  |
| --- |
| 1. In line 8, the BMI formula implemented is: BMI = weight in kg / height in m, while the correct formula for BMI is: BMI = weight in kg / [height in m]^2. The current implementation does not square the height.  bmi = weight / height  This can be corrected by squaring the variable “height” in the formula.  bmi = weight / (height \*\* 2)  2. In line 15, the BMR formula is correct, however the “height” variable is in a unit of meters as per line 6’s input instructions, while the BMR formula requires the height to be in centimeters.  bmr = 10 \* weight + 6.25 \* height - 5 \* age + 5  This can be corrected via multiplying height by 100 in the formula.  bmr = 10 \* weight + 6.25 \* (height \* 100) - 5 \* age + 5 |

Note:

BMR formula for men: 10 \* weight in kg + 6.25 \* height in cm – 5 \* age + 5

BMR for woman: 10 \* weight in kg + 6.25 \* height in cm – 5 \* age -16

Trace your program and complete the trace table again.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Line Number** | **Variable**  **weight** | **Variable**  **height** | **Variable**  **bmi** | **Variable**  **age** | **Variable**  **bmr** |
| 5 | 55 | - | - | - | - |
| 6 | 55 | 1.7 | - | - | - |
| 8 | 55 | 1.7 | 19.031141868512112 | - | - |
| 13 | 55 | 1.7 | 19.031141868512112 | 40 | - |
| 15 | 55 | 1.7 | 19.031141868512112 | 40 | 1417.5 |
|  |  |  |  |  |  |

Activity 8 (InterestCalculator.py)

Open the file “InterestCalculator.py” that you have downloaded from POLITEMall in IDLE.

Given the principal in dollars, rate and duration in years, the formula for calculation of interest is as follows:

*interest = principal × (rate ÷ 100) × duration*

Debug the program using the IDLE Debugger and complete the trace table below after you have corrected the syntax errors:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Line Number** | **Variable**  **principal** | **Variable**  **rate** | **Variable**  **durations** | **Variable**  **interest** |
| 6 | 10000 | - | - | - |
| 7 | 10000 | 10 | - | - |
| 8 | 10000 | 10 | 2 | - |
| 11 | 10000 | 10 | 2 | 200000.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

State the logic error and your correction below:

|  |
| --- |
| 1. Line 11’s interest calculation formula implementation is incorrect as the “rate” is not divided by 100. The correct formula is: Interest = principal \* (rate / 100) \* duration. However, the implemented formula is: Interest = principal \* rate \* duration.  interest = principal \* rate \* duration  This can be corrected by dividing the variable “rate” by 100.  interest = principal \* (rate / 100) \* duration |

Trace your program and complete the trace table again.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Line Number** | **Variable**  **principal** | **Variable**  **rate** | **Variable**  **durations** | **Variable**  **interest** |
| 6 | 10000 | - | - | - |
| 7 | 10000 | 10 | - | - |
| 8 | 10000 | 10 | 2 | - |
| 11 | 10000 | 10 | 2 | 2000.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Activity 9 (StudentData.py)

Create a text file, StudentData.txt, that contains 5 of your friends’ names together with their mobile numbers as shown in screenshot below.

|  |
| --- |
| …. |

Write a program that reads all the data from the file and displays only the first, third and fifth data on the screen.

**OPTIONAL ACTIVITIES**

Question 1 (ChickenRice.py)

Suppose you are working at a chicken rice stall in a hawker centre and you want to calculate the total cost of a plate of chicken rice with any additional side-dishes and GST. Cost for each side-dish is $1.20 and GST is 8%.

Write a Python program to do the following:

* Prompt the user to enter the price of chicken rice
* Prompt the user to enter the number of additional side-dishes
* Calculate the total cost of the purchase together with GST, rounding up to the nearest ten-cent
* Display the results as shown in the sample output below

Sample output (values underlined are the user input):

|  |
| --- |
| Enter the price of chicken rice: 3.5  Enter the number of side-dishes: 1  Total cost of the purchase is $5.10 |

Question 2 (EngineeringComputation.py)

A civil engineering company needs a program that calculates the volume of a cylinder, the surface area of a sphere, and the force of gravity between two objects. The program should take user input for the necessary parameters, and output the results in a formatted output.

Write a Python program to do the following:

* Prompt the user to enter the radius and height of a cylinder.
* Calculate the volume of the cylinder using the formula

where is the constant pi, is the radius of the cylinder and is the height of the cylinder.

* Prompt the user to enter the radius of a sphere.
* Calculate the surface area of the sphere using the formula

where is the radius of the sphere

* Prompt the user to enter the masses of two objects and the distance between them.
* Calculate the force of gravity between the two objects using the formula

where *G* is the gravitational constant ( )

*m1* and *m2* are the masses of the two objects

*r* is the distance between the two objects.

* Display the results as shown in the sample output below

Sample output (values underlined are the user inoput):

|  |
| --- |
| Enter the radius of the cylinder (in m): 2  Enter the height of the cylinder (in m): 5  Enter the radius of the sphere (in m): 3  Enter the mass of object 1 (in kg): 1000  Enter the mass of object 2 (in kg): 2000  Enter the distance between the objects (in m): 10  Results:  Volume of the cylinder: 62.83 cubic meters  Surface area of the sphere: 113.10 square meters  Force of gravity between the objects: 1.33e-06 Newtons |