|  |  |
| --- | --- |
| **Programming 1 (PRG1)**  Diploma in IT / DS / CSF / IM / CICTP  Year 1 (2024/25) Semester 1 | Week **7** |
|  |
| **Exercise 7: For Structure** | |

**OBJECTIVES**

At the end of this exercise, students should be able to develop Python programs using:

* For statement

**IMPORTANT**

* Create a folder, **Week07**, in your hard disk.
* For each question, you will be creating a Python program with the given file name in the **Week07** 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, compress all the files in your **Week07** folder and submit the zip file in POLITEMall

**Activity 1**

Math TimesTable - ( file name: TimesTable.py )

* Write a program that prompts the user to enter a number, and prints out the math times table as shown in the sample output. You are required to use a **for loop**.
* Sample output (input value is underlined):

|  |
| --- |
| Please enter a number: 5  5 x 1 = 5  5 x 2 = 10  5 x 3 = 15  5 x 4 = 20  5 x 5 = 25  5 x 6 = 30  5 x 7 = 35  5 x 8 = 40  5 x 9 = 45  5 x 10 = 50  The End |

*Note: This problem is same as Week 6 Exercise but you are required to use for loop here!*

**Activity 2**

Unique and Odd - ( file name: UniqueOdd.py )

* Given the list of integers as follows:

numbers = [2, 7, 11, 6, 7, 3, 17, 7, 12, 41, 8, 11, 13, 10, 15]

* Write a program that uses **for loop** to read through the list of integers given and
* Chooses only the first 5 unique odd numbers it encounters.
* Discard the duplicates that it finds.
* Creates a list to store the findings.
* Prints the 5 unique odd numbers found.
* If there are fewer than 5 unique odd numbers found, the program will print all the unique odd numbers available.
* Sample output:

|  |
| --- |
| [7, 11, 3, 17, 41] |

**Activity 3**

Keeping track of progress - ( file name: TaskList.py )

* Tom would like to create a task list dynamically according to the number of days.
* Write a Python program that prompts Tom to enter the number of days, and print the day number in the first column, and spaces for him to fill up his task in second column. After every 7 days, there should be a heading inserted.
* Sample output (input value is underlined):

|  |
| --- |
| Enter number of days: 9  Day | Task(s)  1 |  2 |  3 |  4 |  5 |  6 |  7 |  Day | Task(s)  8 |  9 | |

**Activity 4**

Odd or Even - ( file name: OddEven.py )

* Write a program that allows the user to enter several integers until a zero is entered. The program upon receiving a zero “0” will then proceed to list the following:
* The odd integers in increasing order.
* The even integers in increasing order. (NOTE: Zero is not part of the list)
* The average of all the numbers.
* Sample output (input value is underlined):

|  |
| --- |
| Please enter a number (0 to end): 14  Please enter a number (0 to end): -23  Please enter a number (0 to end): -8  Please enter a number (0 to end): 72  Please enter a number (0 to end): 51  Please enter a number (0 to end): 27  Please enter a number (0 to end): 18  Please enter a number (0 to end): 0  Odd numbers: [-23, 27, 51]  Even numbers: [-8, 14, 18, 72]  Average = 21.57 |

**Activity 5**

Test Scores - ( file name: Scores.py )

* The file “scores.txt” contains a list of students’ id, name and scores for two tests as shown below.

A close up of a number

Description automatically generated

* Write a program to read the data from the file, calculate and print the students’ info as well as the average of two test scores. Find and display also the student with the highest average.
* Sample output:

|  |
| --- |
| ID Name Test1 Test2 Average  47683985V David 41 56 48.50  80587257L Grace 55 76 65.50  47295164J Alice 80 75 77.50  39180601A Charlie 78 67 72.50  97401326O Jack 100 99 99.50  96247157K Henry 88 90 89.00  23389267N Frank 45 34 39.50  29439749G Bob 65 60 62.50  21915630I Emma 69 70 69.50  82939283X Ivy 20 30 25.00  Jack has the highest average score of 99.5 |