|  |  |
| --- | --- |
| **Programming 1 (PRG1)**  Diploma in IT / DS / CSF / IM / CICTP  Year 1 (2025/26) Semester 1 | Week **6** |
|  |
| **Exercise 6: While Structure** | |

**OBJECTIVES**

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

* while statement
* flow control

**IMPORTANT**

* Create a folder, **Week06**, in your hard disk.
* For programming questions, create Python programs with the given file names in the **Week06** folder created above. Do add the description, your name and student ID as comments at the beginning of each program.
* For non-programming questions, type your answers in the boxes provided below the questions.
* At the end of the session, compress all the files in your **Week06** folder (i.e. the Python program files and this word document) 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 while 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 |

**Activity 2**

Count Pushups - ( file name: PushupCounter.py )

* Write a program that asks the user for a target number of pushups. It then asks the user to enter the number of pushups he has done each day and continues to do so until he has reached his target. The program will then show the total number of pushups as well as the number of days the user took to reach the target. You are required to use a while loop.
* Sample output (input value is underlined):

|  |
| --- |
| Enter target number of pushups: 100  Day 1: How many pushups did you do today? 28  Day 2: How many pushups did you do today? 27  Day 3: How many pushups did you do today? 30  Day 4: How many pushups did you do today? 31  You did a total of 116 pushups.  You hit your target in 4 days! |

**Activity 3**

Check Pin - ( file name: GetPin.py )

* Write a program to prompt the user for pin until correct pin has been entered. If the user fails to enter the correct pin after 3 times, lock the account. You may assume that the correct pin is ‘12345’. You are required to use a while loop.
* Sample output (input value is underlined):

|  |
| --- |
| Enter pin: 123  Invalid pin. Please try again.  You have 2 tries left.  Enter pin: 234  Invalid pin. Please try again.  You have 1 tries left.  Enter pin: 1234  Invalid pin. You have no more tries.  Your account is locked. |

|  |
| --- |
| Enter pin: 00000  Invalid pin. Please try again.  You have 2 tries left.  Enter pin: 12345  Correct pin entered. |

**Activity 4**

Calculate Average Temperature – ( file name: TemperatureSensor.py )

* A room is installed with a sensor that measures the room temperature at an hourly interval. The temperatures are stored in the text file “temperature.txt”.
* Write a program to do the following:
  + read the data from the text file and store the values in a list;
  + display all the data with a warning message if the temperature is above 29 degrees;
  + calculate and display the average temperature.
* You are required to use a while loop.
* Sample output:

|  |
| --- |
| The temperatures are  20.5  22.0  21.0  29.3 \*\* higher than 29!!!  28.2  25.0  26.0  28.0  26.3  25.6  29.3 \*\* higher than 29!!!  28.4  24.5  26.3  25.5  26.5  23.3  24.3  25.4  26.5  23.0  3.0  25.4  26.3  25.5  Number of readings: 25  Average temperature: 24.60 |

**Activity 5**

Determine if the following code will result in infinite loop:

|  |  |
| --- | --- |
| while not(True): print() |  |
| while 0: print() |  |
| count = '0'  while count!=0: print() |  |
| count = 0  while count>=0: print() |  |

**Activity 6**

Counting Words – ( file name: WordCount.py )

* Write a program that adds up to 5 unique words to a list. The user can enter 'x' to stop adding words. Once either 5 words are added or the user stops the program early, the words are listed and the total number of letters in the words are displayed.
* Sample output (input value is underlined):

|  |
| --- |
| Enter word (x to exit): Programming  Enter word (x to exit): x  Your words are ['Programming']  The number of letters in these words is 11 |

|  |
| --- |
| Enter word (x to exit): roses  Enter word (x to exit): are  Enter word (x to exit): red  Enter word (x to exit): violets  Enter word (x to exit): are  are has already been entered.  Enter word (x to exit): blue  Your words are ['roses', 'are', 'red', 'violets', 'blue']  The number of letters in these words is 22 |

**Activity 7**

Number Guessing Game – ( file name: NumberGuessing.py )

* Write a program that simulates a number guessing game. It first generates a random number between 1 and 100. It then prompts user to guess the correct number. User can enter -1 to end the game or the game will end after 5 tries.
* Do print out the correct number if the user did not get it right.
* Sample output (input value is underlined):

|  |
| --- |
| Welcome to Number Guessing Game  Try 1: Enter a number between 1 and 100 (or -1 to end): 50  50 is too low.  Try 2: Guess again, enter a number between 1 and 100 (or -1 to end): 75  75 is too high.  Try 3: Guess again, enter a number between 1 and 100 (or -1 to end): 63  63 is too low.  Try 4: Guess again, enter a number between 1 and 100 (or -1 to end): 64  Bingo, you've got it right!  Bye-bye! |

**OPTIONAL**

**Activity 8**

Student marks – ( file name: Marks.py )

* The text file “marks.txt” contains some students’ names and their individual mark for a test separated by ‘;’ as follows:

|  |
| --- |
| David Lim;80  Peter Tan;90  Asli Fan;67  Simon Chan;77 |

* Write a Python program to read the data from the file, print each student’s name and mark as well as the average mark for all students as shown in sample output below.

|  |
| --- |
| Name Mark  ---- ----  David Lim 80.0  Peter Tan 90.0  Asli Fan 67.0  Simon Chan 77.0  Average Mark: 78.50 |

Hint: You may use readlines() to read all data in the file and store in a list