|  |  |
| --- | --- |
| **Programming 1 (PRG1)**  Diploma in IT / DS / CSF / IM / CICTP  Year 1 (2023/24) Semester 1 | Week **11** |
|  |
| **Exercise 8: Nested Loops** | |

**OBJECTIVES**

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

* Nested loops

**IMPORTANT**

* Create a folder, **Week11**, in your hard disk.
* For each question, you will be creating a Python program with the given file name in the **Week11** 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 **Week11** folder and submit the zip file in POLITEMall.

**Activity 1**

Christmas Tree – (file name: ChristmasTree.py)

* Write a Python program to prompt the user for a character and a number, and then print a Christmas tree accordingly.
* Sample output:

|  |
| --- |
| Enter a character: \*  Enter a number: 6  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  \* \* \* \* \* \*  Merry Christmas! |

**Activity 2**

Print Map – (file name: PrintMap.py)

* The map for a computer game is stored in the following nested list:

map = [ ['T', ' ', ' ', ' ', 'T'],\

[' ', 'P', ' ', ' ', ' '],\

[' ', ' ', ' ', 'T', ' '],\

[' ', 'T', ' ', ' ', ' ']

]

* Write a Python program that prints the map below:
* Sample Output:

|  |
| --- |
| +---+---+---+---+---+  | T | | | | T |  +---+---+---+---+---+  | | P | | | |  +---+---+---+---+---+  | | | | T | |  +---+---+---+---+---+  | | T | | | |  +---+---+---+---+---+ |

**Activity 3**

Print Calendar – (file name: PrintCalendar.py)

* Write a Python program to print the month view of a calendar. Your program should do the following:
  + prompt the user for the number of days in the month and which day of the week does the first day of the month falls on
  + display the calendar as shown in the sample output below
* Sample Output:

|  |
| --- |
| Enter number of days: 31  When is the first day of the week: Tue  Sun Mon Tue Wed Thu Fri Sat  1 2 3 4 5  6 7 8 9 10 11 12  13 14 15 16 17 18 19  20 21 22 23 24 25 26  27 28 29 30 31 |

**Activity 4**

Video Game Competition – (file name: GameScores.py)

* Six gamers are playing in a video game competition. They play several games and score points for each game. If the points scored for a game is 100 or more, it is a win, otherwise it is a loss. The names of the players are stored in the following list:

player = ['Hafu', 'Toast', 'Pokimane’,

'Pewdiepie', 'Ninja', 'Markiplier’]

* The results of the games played so far by each gamer is given by the following list:

results = [ [98, 107, 87, 121],

[88, 111],

[79, 130, 99],

[86, 100, 121, 66, 98],

[108, 79, 92],

[77, 126, 93, 100, 73, 89],

]

* Write a Python program to display the details of the games played by each player as shown in sample output below.
* Sample Output:

|  |
| --- |
| Player Games Wins Total  Hafu 4 2 413  Toast 2 1 199  Pokimane 3 1 308  Pewdiepie 5 2 471  Ninja 3 1 279  Markiplier 6 2 558 |

* Column **Games** is number of games played by the player
* Column **Wins** is the total number of wins achieved
* Column **Total** is the total number of points the player has so far

**OPTIONAL**

**Activity 5**

Simple Digital Pet – (file name: Digipet.py)

Being a big fan of Pokémon, Tom wants to create his own digital pet. In the digital pet, there are options to check the status of the pet, feed the pet, play with the pet, or let the pet rests. In the status section, player can see the pet’s level of hungriness, happiness, and energy. Each status can have a maximum level of 5 stars indicating full, or minimum 5 dots if the level is empty. If the player feeds the pet, the hungriness level will go up by 1 star, meanwhile causing the other 2 status to drop by 1 star. If the player plays with the pet, the happiness will go up by 1 star, while the other 2 will go down by 1 star. Likewise, if the player let the pet rests, the energy goes up by 1 star, and the other 2 indicators will drop by 1 star.

Write a Python program that allows the player to interact with the digital pet as described above.

You may define the 4 lists below at the beginning of your program:

menu = ['Feed','Play','Rest','Status']

status = [3,3,3]

title = ['hungry','happiness','energy']

msg = ['Nom nom nom','XD','Zzzzz']

Sample output:

 