Week 5: Loops

Programming

2025 – 2026

Bachelor in Data Science and Engineering

Dual Bachelor in Data Science and Engineering and Telecommunication Technologies Engineering

Exercise 1. Create a program that generates a random number between 1 and 20, do not show this number on the screen and keep it hidden. Then try to find out its value. As you enter values the program must indicate if the number entered is greater or less than the one that the program has generated and also the number of tries.

Exercise 2. Ask the user two integer numbers A and B such as A + 5 < B. Keep asking until the former condition is met. Then randomly generate and print 5 integer numbers in the interval [A,B] such as they are even and odd in an alternative way. It does not matter if the values are repeated; the aim is alternatively printing odd and even numbers.

Example: for the interval [3,9] a valid sequence of numbers is: 6, 7, 6, 3, 4

Exercise 3. Create a program to generate and print on screen the N perfect numbers smaller than the value that the user introduces using the keyboard. A number is a perfect number when it is equal to the addition of all its divisors except itself (You can rely on the flow diagram of the similar exercise of a previous week). Example:

```
Introduce the top limit to generate perfect numbers and press Enter
10000
The number 6 is perfect
The number 28 is perfect
The number 496 is perfect
The number 8128 is perfect
```

Exercise 4. As you know, when we ask the user to enter a number in Python using `input`, we need to cast the number to `int` or `float` to be able to work with it. But if the user enters something which is not a number the program fails. A possible solution is to use the `isdigit()` function of strings, which returns `True` if the string is a number (for example `'3434'.isdigit()` returns `True`). This works for integer numbers but not for `float`, as the point is not recognized as a number (`'34.22'.isdigit()` returns `False`). Create a program that asks the user to enter a number and keeps asking until a number is introduced. At the end it prints the square of the entered number. It must work both for integer and float numbers.

Exercise 5. Define a program that allows simulating the behavior of an ATM. The program must meet the following requirements

- Randomly create a 4-digits PIN number and the account balance, which must be in the range 50 to 5,000 Euros. Notice that the PIN must be stored as string, otherwise any leading zero it may have will be lost.

- Request the corresponding PIN code to the user, giving up to three attempts to enter the correct number. In case of failing three times the program will show a message on the screen and finish.

- If the pin is correct it must show a menu with the different operations that the ATM allows, as seen in the next example:

```
Welcome
------------------------
1- Deposit
2- Cash withdrawal
3- Exit
Choose the operation:
```

After each operation, indicate on the screen the available balance in the account. If the cash withdrawal operation exceeds the amount available in the bank account, the operation will be denied and the corresponding message will be displayed.

Exercise 6. Define a program which requests the mark of the exam for each student in a class and calculates the highest mark, the lowest, the average and the number of students attending to the exam. The data input must finish when a negative number is introduced.

Exercise 7. Create a program which plays "Rock, paper, scissors", with the next specifications:

- There will be one player who plays against the machine.
- The number of games will be a constant in the program. The game will be repeated the number of times indicated in this constant.
- The following message will be shown in order to ask the user for the move:

  ROCK, PAPER OR SCISSORS?

  The program will randomly obtain the move (rock, paper or scissors). The program will show the winner according to the following rules:
  - Rock crushes scissors (Rock wins)
  - Scissors cuts paper (Scissors wins)
  - Paper wraps stone (Paper wins)
  - In case of tie, the message “TIE” will be shown.

EXAMPLE OF PROGRAM EXECUTION

```
***** 4 games will be run
ROCK, PAPER or SCISSORS? PAPER
Program chooses rock
*****PLAYER WINS*****
ROCK, PAPER or SCISSORS? ROCK
Program chooses paper
*****PROGRAM WINS*****
ROCK, PAPER or SCISSORS? SCISSORS
Program chooses rock
*****PROGRAM WINS*****
ROCK, PAPER or SCISSORS? ROCK
Program chooses rock
*****TIE*****
```

Exercise 8. Create a program that plays "Twenty 0ne", a simple dice version of Blackjack. For this game two dices will be rolled. The points will be calculated as the sum of the rolls of the two dices. Although the game can be played with several players, to simplify it we will play only with two.

The rules are as follows:

- Each player plays only once per game.
- The player can roll the dice as many times as desired and can stop at any time if the score is no higher than 21. Numbers obtained in each roll are added to the previous in order to establish the score.
- If a player reaches a score greater than 21 she loses immediately, so the other player is automatically the winner.
- If no player has exceeded 21, the participant whose total is nearest to 21 points, wins. If the two players reach the same score, then it will be declared a tie.
- The number of games to be played will be a value defined as a constant in the program.

Taking into account the specifications established and the example provided:

- Program the move corresponding to player 1 until she decides to stop, or the score is higher than 21, in this last case player 1 has lost.
- Program also the move corresponding to player 2 and display the winner (or the tie)
- Modify the exercise to run multiple games as indicated in the statement.

EXAMPLE OF PROGRAM EXECUTION

```
GAME 1 - PLAYER 1
The number of points obtained are 6, 5
The points accumulated are 11
Would you like to roll the dice again? (yes/no) yes
GAME 1 - PLAYER 1
The number of points obtained are 3, 5
The points accumulated are 19
Would you like to roll the dice again? (yes/no) no
GAME 1 - PLAYER 2
The number of points obtained are 3, 5
The points accumulated are 8
Would you like to roll the dice again? (yes/no) yes
GAME 1 - PLAYER 2
The number of points obtained are 5, 4
The points accumulated are 17
Would you like to roll the dice again? (yes/no) no
***********PLAYER 1 WINS***********
```

Delivery rules

The solutions to the previous exercises must be uploaded to Aula Global. Upload a zip file containing a file for each exercise (name them `exercise1.py`, `exercise2.py`, etc). The name of the zip file must be "lab05-name-initials.zip" (Lucía Pérez Gómez will name the file lab05-lpg.zip).