

Lab 1 — Python Fundamentals

Prof. Carlos J. Costa

Objective

This laboratory introduces fundamental Python programming concepts:

- Variables and data types
- Input and output
- Arithmetic operators
- Comparison and logical operators
- Conditional statements
- String manipulation
- Mathematical functions
- Modules and libraries
- Basic financial calculations

Instructions: Write and test a Python solution for each exercise. Unless explicitly requested, do not use external libraries.

1. Check if a Value Is an Integer

Ask the user to enter a value and verify whether it is an integer.

Example:

Enter a value: 15

15 is an integer.

2. Check if a Number Is Even

Ask the user to enter an integer and verify whether the number is even or odd.

Example:

Enter an integer: 24

24 is even.

Hint: Use the modulo operator %.

3. Compare Two Numbers

Ask the user to enter two numbers.

Determine whether the first number is:

- greater than the second number;
- smaller than the second number; or
- equal to the second number.

Example:

First number: 15

Second number: 10

15 is greater than 10.

4. Check if a Number Is a Multiple of Another

Ask the user to enter two integers.

Determine whether the first number is a multiple of the second number.

Example:

First number: 20

Second number: 5

20 is a multiple of 5.

Hint: Use the modulo operator %.

5. Calculate Simple Interest

An investor deposits **€200** for **3 years** at an annual interest rate of **3%**.

Calculate the interest earned using:

$$I = P \times R \times T$$

where:

- I = interest
- P = principal (initial capital)
- R = interest rate
- T = time in years

Important: Convert the interest rate from a percentage to a decimal before performing the calculation.

Display the interest earned.

6. Calculate Compound Interest

An investor deposits **€200** for **3 years** at an annual interest rate of **3%**.

Calculate the final capital using compound interest:

$$A = P(1 + R)^T$$

where:

- A = final capital
- P = initial capital
- R = annual interest rate
- T = number of years

Display the final capital and the total interest earned.

7. Calculate BMI

Ask the user to enter their:

- mass in kilograms;
- height in metres.

Calculate the Body Mass Index (BMI):

$$BMI = \frac{mass}{height^2}$$

Display the calculated BMI.

Example:

Mass (kg): 70

Height (m): 1.75

BMI: 22.86

8. Calculate the Golden Ratio

Calculate the Golden Ratio:

$$\varphi = \frac{1 + \sqrt{5}}{2}$$

Solve the problem in **two different ways**.

a) Without importing a mathematical library

Use Python's available operators/functions to calculate the square root.

b) Using the math module

Import the math module and use:

```
import math
```

```
math.sqrt()
```

Compare the two solutions.

9. Calculate the Net Present Value (NPV)

An investor makes an initial investment of **€10,000**.

The expected cash inflows are:

Year Cash Flow

1 €2,000

2 €3,000

3 €4,000

4 €4,000

5 €5,000

The discount rate is **10%**.

Calculate the Net Present Value (NPV):

$$NPV = \sum_{t=1}^n \frac{CF_t}{(1+k)^t} - I$$

where:

- CF_t = cash flow in year t
- I = initial investment
- k = discount rate
- t = year

Display the NPV.

Question: Based only on the NPV, would the investment be financially attractive?

10. Calculate a Person's Year of Birth

Ask the user to enter:

- their name;
- their age.

Calculate the person's approximate year of birth and display a message such as:

Carlos was born approximately in 2000.

Solve the problem in **two ways**.

a) Without using modules

Use a fixed reference year.

b) Using the datetime module

Use:

```
from datetime import date
```

```
today = date.today()
```

```
today.year
```

Use the current year to calculate the birth year.

Question: Why can the result differ by one year depending on whether the person's birthday has already occurred?

11. Manipulate a Person's Name

Ask the user to enter their **forename(s)** and **surname(s)**.

Create a variable called `name` containing the complete name.

Then create the following variables:

nameBig

The complete name in uppercase.

nameTitle

The first character of each word is uppercase and the remaining characters are lowercase.

nameSmall

The complete name in lowercase.

nameCapitalized

Only the first character of the complete name is uppercase.

Example:

For:

Forename: carlos

Surname: COSTA

the results should be equivalent to:

`name` → carlos COSTA

`nameBig` → CARLOS COSTA

`nameTitle` → Carlos Costa

`nameSmall` → carlos costa

`nameCapitalized` → Carlos costa

Hint: Investigate the following string methods:

`.upper()`

`.lower()`

`.title()`

`.capitalize()`

12. Find a Substring in a String

Create a string containing a person's complete name.

Use the following method to determine the position of the **first occurrence** of "da":

`str.find(sub, start, end)`

For example:

`name.find("da")`

Answer the following questions:

1. What position does `find()` return when "da" is found?
2. What does `find()` return when "da" is not found?
3. Modify your program so that it displays an appropriate message in both cases.

Hint: Python string positions start at **0**.

Submission

For each exercise:

1. Write the Python code.
2. Execute the code with at least one test case.
3. Check that the result is correct.
4. Add comments where they help explain your solution.

Do not simply provide the final numerical answer: **the objective is to develop and test the Python program.**