

CS1205 Programming Fundamentals I

Rosane Minghim

November.2025

Assignment 1 - 10 marks

To be submitted via Canvas by Nov. 15th

General Instructions

1. Your assignment is to be developed in Python. It should run on Spyder Notebook.
2. Your code should be commented as to what each set of commands is doing. Your functions should be commented as we have been doing in class. These comments will be considered in the final grade, in that there will be discounts caused by missing comments on important parts (.5 mark per lack of comment, up to 3 marks). *The first comment at the top of your file should must be your name and student number.* Do not comment in excess.
3. All outputs (`print` commands) should have proper messages and be formatted adequately. **No print should be inside your functions in the submitted version of your assignment. All printing is done in the main program.**
4. **The submitted version of this assignment should have no input from the user. All variables to test your functions should be pre-set in your program.** There should be no use of the `input()` function inside functions. All inputs in functions are via parameters only.
5. Make sure none of your assignment code is ever shared or left for others to see. Plagiarism will be verified.
6. You should submit a comment at the start of the code for each question (ex. `# Question 1, # Question 2, etc.`). Leave at least two blank lines before and after the solution for each question.
7. All your answers should be added to a single Python file and submitted to Canvas. If Canvas does not allow Python '.py' submission you must 'zip' the code and submit as a zip file. Please test in advance, as deadlines cannot be extended. You will be able to submit as many versions as necessary until the deadline. I can only see the last one, so make sure the version submitted by the deadline is the one to be graded.
8. Plagiarism penalties apply.

Assignment Specifications

1. **BMI** (Body Mass Index) is a number that can be calculated from someone's height (in meters) and weight (in kg), and will give as a result a value, whose range indicates a weight classification (for information, see Table 1).

The formula for BMI is given by:

$$BMI = \text{weight}(kg)/\text{height}(meters)^2$$

Define **YOUR OWN** Python function that has as parameters a user a **height** value (floating point, in meters) and **weight** (floating point, in kilograms), and **returns** the corresponding BMI.

Write code in your main program to properly test your function with preset variables in the main program.

2. Many societies use Imperial Scales, rather than metric scales. Define your own two Python **functions** to:
 - (a) Convert height (in *Imperial scale*) to metric. This function takes the number of feet and the number of inches as parameters and returns the height in meters (floating-point).
 - (b) Convert weight (in *Imperial scale*) to metric. This function takes the number of stones and the number of pounds (both integers) as parameters and returns the weight in kilograms (floating-point).

Test your conversion functions properly in your code, using pre-set variables in the main program.

Obs:

- (a) to convert feet to meters (approximately), **divide** the value in feet by 3.281.
 - (b) to convert inches to meters (approximately), **divide** the value in inches by 39.37.
 - (c) to convert stones to kg (approximately), multiply the value in stones by 6.35.
 - (d) to convert pounds to kg (approximately), divide the value in pounds by 2.205.
3. Write a Python function that receives a value of BMI as the parameter (floating point) and returns a string for the weight classification corresponding to that BMI value, according to Table 1.

Test your function properly with values in all ranges and on the borders of the ranges. Use only pre-set variables in your program in your submitted code. Remember: no input from the user.

Print all results in your main program.

4. Finally, define and test a function in Python that takes the height and weight of a person (in Imperial scale) and returns the weight classification of that person, BY CALLING YOUR FUNCTIONS ABOVE to convert height and weight to metric, calculate BMI and give weight classification as a result.

Example: Say your function prototype is:

patient_weight_classification (feet, inches, stones, pounds).

- If you test it by calling:
`print(patient_weight_classification(5,10,13,0))`
you would get:
'overweight'
as the corresponding BMI for a person 5 feet 10 inches and 13 stones is 26.
- If you test it by calling:
`print(patient_weight_classification(6,1,12,3))`
you would get:
'healthy weight'
as the corresponding BMI for a person 6 feet 1 inch and 12 stones 3 pounds is 22.6.

Table 1: BMI levels for reference

BMI Levels and Weight Classification		
Level	Weight Range	Weight Classification
1	< 18.5	underweight
2	18.5–24.9	healthy weight
3	25.0–29.9	overweight
4	30.0–34.9	obesity I
5	35.0–39.9	obesity II
5	≥40	obesity III