

BLOCK NAME	SOFTWARE DEVELOPMENT FUNDAMENTALS II
BLOCK CODE	CS-L1B2
COURSE	1
LEVEL	1
CREDITS	6
CLASS HOURS	60
HOMEWORK	90
TOTAL HOURS	150

DESCRIPTION

This block delves into software development with a view of the internal structure and operation of computers. We face the challenge of solving a practical case for which we will need to use basic built-in memory structures and work with files, that in turn will force us to be able to debug our source code. We also face a challenge for which we will need to create a little assembly program, in order to go deeper into the operation of computers.

PRE-REQUISITES

Basic programming skills are needed.
CS-L1B1

OBJECTIVES

The goal is for students to understand how a computer works in order to take correct decisions when choosing the way of solving specific problems.

SKILLS TO BE DEVELOPED

- 1 - Basic memory structures.**
 - 1.1 - Use python strings.
 - 1.2 - Use python lists.
- 2 - Files.**
 - 2.1 - Open and read files.
 - 2.2 - Write output to file.
- 3 - CPU.**
 - 3.1 - Understand how a CPU works (ALU, registers, memory access).
 - 3.2 - Know main instruction set types (CISC and RISC).
 - 3.3 - Understand why different instructions need different amounts of CPU cycles.
 - 3.4 - Understand how microprocessor frequency affects performance.
- 4 - RAM memory.**
 - 4.1 - Understand the Cache/RAM hierarchy.
 - 4.2 - Understand how virtual memory works.
- 5 - Permanent storage.**
 - 5.1 - Understand how permanent storage works.
 - 5.2 - Differences between using HDDs and using SSDs.
- 6 - Global lessons about CPU, RAM and storage.**
 - 6.1 - Understand the IPC (Instructions per Clock) concept.
 - 6.1 - Understand the differences between accessing RAM memory and accessing storage.
- 7 - Basic program debugging.**
 - 7.1 - Understand the drawbacks of screen messages for debugging.
 - 7.2 - Use log files for debugging.
 - 7.3 - Use pdb for debugging.

SYLLABUS

- 1 - Basic memory structures.
- 2 - Files.
- 3 - CPU.
- 4 - RAM memory.
- 5 - Permanent storage.
- 6 - Basic program debugging.

METHODOLOGY

Resolution of practical activities supervised by the mentor. Compulsory attendance.

DEDICATION AND EVALUATION

The student must pass the mandatory activities (challenges/projects) that are covered in the block.

Each challenge/project produces its own score and has been designed to cover certain block percentages.

Such score is 80% objective (the program that solves the challenge/project works without errors and producing the expected results) and 20% subjective (solution elegance, how clean the code is, documentation).

Block scores are finally calculated by prorating individual activities with respect to their block coverage percentages.