IMMUNE TECHNOLOGY INSTITUTE

**COMPUTER ENTREPRENEURSHIP BACHELOR**

| | |
|---|---|
| **BLOCK NAME** | **SOFTWARE DEVELOPMENT FUNDAMENTALS I** |
| **BLOCK CODE** | **CS-L1B1** |

| | |
|---|---|
| **COURSE** | **1** |
| **LEVEL** | **1** |
| **CREDITS** | **6** |
| **CLASS HOURS** | **60** |
| **HOMEWORK** | **90** |
| **TOTAL HOURS** | **150** |

## DESCRIPTION

This block introduces the fundamentals of software development, enabling the student to start creating basic desktop programs. We start installing a Ubuntu distribution on our laptop and learning how to use Ubuntu at user level. Then we follow the official Python Tutorial to learn the basis of programming and finally face the challenge of solving a practical case for which we will need to make use of what we have just learned.

## PRE-REQUISITES

The student should be familiar with the basic usage of personal computers. No previous knowledge about software development is needed.

## OBJECTIVES

The goal is for students to acquire basic programming knowledge that will accompany them while developing any kind of software.

## SKILLS TO BE DEVELOPED

**1 - Linux.**
1.1 - Install a Ubuntu distribution from scratch.
1.2 - Know how Ubuntu works.
1.3 - Find and use the main Ubuntu applications.
1.4 - Create, rename, move and delete folders.
1.4 - Create, rename, move, edit and delete files.
**2 - Creating basic programs.**
2.1 - Create a basic Python3 program and execute it.
**3 - Variables.**
3.1 - Define and use variables.
**4 - Control structures.**
4.1 - Define and use control structures.
**5 - Conditions.**
5.1 - Write complex conditions for your control structures.
**6 - Functions.**
6.1 - Write and call functions with parameters and return value.
**7 - Input/Output.**
7.1 - Make your program receive user input from keyboard.
7.2 - Make your program output information to screen.
**8 - Built-in data structures.**
8.1 - Use basic built-in python data structures.
**9 - Exceptions.**
9.1 - Make use of exceptions to manage errors.

## SYLLABUS

1 - Linux.
2 - Creating basic programs.
3 - Variables.
4 - Control structures.
5 - Conditions.
6 - Functions.
7 - Input/Output.
8 - Built-in data structures.
9 - Exceptions.

## 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.