MEDTEC SHOOL

Course: Computer Science

Year (1st-2nd-3rd-4th-5th-6th): 2nd

Period (1st-2nd semester – annual): 1st

Credits: 5

Objectives

The Computer Science course introduces problem solving through computer programming, with the following objectives: i) to develop a consistent methodology for solving problems, ii) to present the fundamental aspects of the Python programming language, and iii) illustrate the problem solving process with Python through a variety of interesting examples, either framed in the engineering or medical environment.

At the end of the course, students will acquire:

Knowledge and understanding
Students will acquire knowledge on and understanding of basic elements of computer programming and algorithms and will be able to demonstrate:

- *knowledge of how information is stored and handled in a computing system*
- *knowledge of the Python programming language syntax and semantics*
- *knowledge of the main statements typical of imperative programming languages to manipulate data*

Ability to apply knowledge and understanding
Students will use the acquired knowledge to achieve the following goals:

- *select and design data structures to effectively store information*
- *given a simple problem framed in the engineering or medical environment, design algorithm for its efficient resolution*
- *implement the algorithm in Python*

Making judgements
Students will acquire knowledge on how to interpret data aiming at formulating a subjective judgement:

- *evaluate the correctness of the proposed algorithm*
- *evaluate the appropriateness of the proposed algorithm/program for solving the given problem*

Prerequisites

Basic ability to use a personal computer

Contents

The course is organized in lectures and practical sessions, and will cover the following topics:

- *computer architecture basics*
- *information representation*
- *fundamental aspects: structure of a Python program, built-in basic types and arithmetic/logical operators, statements, input/output, basic control structures/statements*
- *complex data types (lists, dictionaries, sets) and built-in methods*
- *functions, parameters and scopes*
- *files*
- *data manipulation and visualization*

Teaching Methods

The course is organized in frontal lectures introducing the key concepts and problem solving strategies using Python and practical sessions where exercises and small projects will be proposed and solved by writing Python programs. Lectures and practical sessions will provide the problem solving and programming abilities and skills that will be verified during the exam sessions.

Students are encouraged to actively participate to the lectures with questions and comments.

Assessment

The exam can be taken in each one of the scheduled sessions in January-February, July and September.

A programming project will be proposed for each exam session, two weeks in advance.

Each student will develop autonomously and independently a program that behaves as requested and will write a short document that simply reports and explains how the program behaves when executed on the available data set, specified in the project description. Students are free to discuss the requirements of the projects, the elements that need clarification, but you cannot discuss and share the solution.

An oral discussion (open to anybody) will be held subsequently, so that the student can explain and motivate the adopted choices and eventually modify her/his own code to improve it or correct mistakes.

The project will be evaluated in terms of the quality of the adopted choices with respect to adopted data structures and solving algorithm and the correctness of the proposed solutions. The oral discussion will contribute to the grade in terms of the ability to explain the presented solution and to adapt it upon request.

The exam is passed upon obtaining a score equal or greater than 18/30, being the maximum score 30/30.

Texts

M. Lutz, "Python Pocket Reference", O'Reilly
J. Hunt, "A Beginners Guide to Python 3 Programming", Springer