Course Plan: Computer Science 1

Dr. Adel Abdelli

adel.abdelli@univ-biskra.dz

1.0

17/03/2024

Table of contents

1. *Information about the course*

Faculty: Science and Technology

Department: Common Core in Science and Technology

Target audience: 1st year bachelor students

Title of the course: Computer Science 1

Credit: 04

coefficient: 02

Duration: 15 weeks

Schedules:Wednesday,9:40-11:10

Room: A4

Teachers:

- Lectures : Dr. Adel ABDELLI

  Contact: adel.abdelli@univ-biskra.dz

Availability :

Answer on the forum: any question related to the course must be posted on the dedicated forum

so that you can all benefit from my response, I undertake to answer the questions posted within 48 hours.

By email: I undertake to respond by email within 48 hours of receipt of the message, except in the event of unforeseen circumstances, I draw your attention to the fact that the preferred communication channel is the forum, the email is reserved for “emergencies” (in the event of a problem accessing the platform) and it must be used with discernment.

2. *Course presentation*

This course is part of the core curriculum of the Science and Technology field. Therefore, it is intended for first-year students and aims to provide them with a general introduction to computer science and coding.

The program first addresses the history of computing, the main computer components and their functioning. It then presents fundamental concepts such as digital systems, algorithms and programming.

In the second part of the semester, students learn the basics of a programming language such as Python. They then practice their knowledge by coding small programs under the supervision of teachers.

The aim of this course is to familiarize students with the world of computing and provide them with the necessary technical and theoretical foundations for the rest of their studies and the job market. Evaluation can be done through continuous assessments and the completion of a programming project at the end of the semester.

3. ***Content***

This the content of the course (check also the map mind below):

Chapter 1: Introduction to Computer

1. Science History of computing.
2. Hardware components (CPU, memory, storage).
3. Software components (operating systems, programming languages) Data representation (binary, hexadecimal).
4. Algorithms and problem solving.

Chapter 2: Notions of Algorithm and Program

1. What is an algorithm.
2. Steps to solve computational problems.
3. Programming fundamentals (variables, conditions, loops).
4. Basic program structure and design.

Chapter 3: Indexed Variables

1. Arrays as indexed variables.
2. Array declaration and initialization Accessing array elements.
3. Common array operations (sorting, searching, manipulation) Multidimensional arrays.
4. Array algorithms (summing, averaging, filtering).

The map mind of the course:

4. *Prerequisites*

In this class, we aim to make the subject matter accessible to everyone, regardless of their background in technology. While having a basic understanding of computers and their components is beneficial, we will ensure that even those who are relatively new to the field can grasp the concepts we cover. We'll start by exploring the fundamentals of computer science, delving into topics such as algorithms, data structures, programming languages, and computational thinking.

Throughout the course, we'll provide ample explanations and examples to aid your understanding, so don't worry if you're not already an expert. Our goal is to equip you with the knowledge and skills needed to navigate the world of computer science confidently.

So, whether you're a seasoned tech enthusiast or just dipping your toes into the realm of computing, you'll find this class to be an enriching and rewarding experience.

To test your prerequisites, a test is made available to students on the distance education platform. You must use the identifiers (username and password) provided by the teacher to log in.

5. *Learning goals*

This course aim to make the student capable to:

1. Recognize the components of the computer.

2. Indicate the differences between the hardware and the software parts of a computer.

3. Understand an algorithm and how to illustrate it.

4. Solve a problem using an algorithm.

5. Interpret an algorithm into Python code.

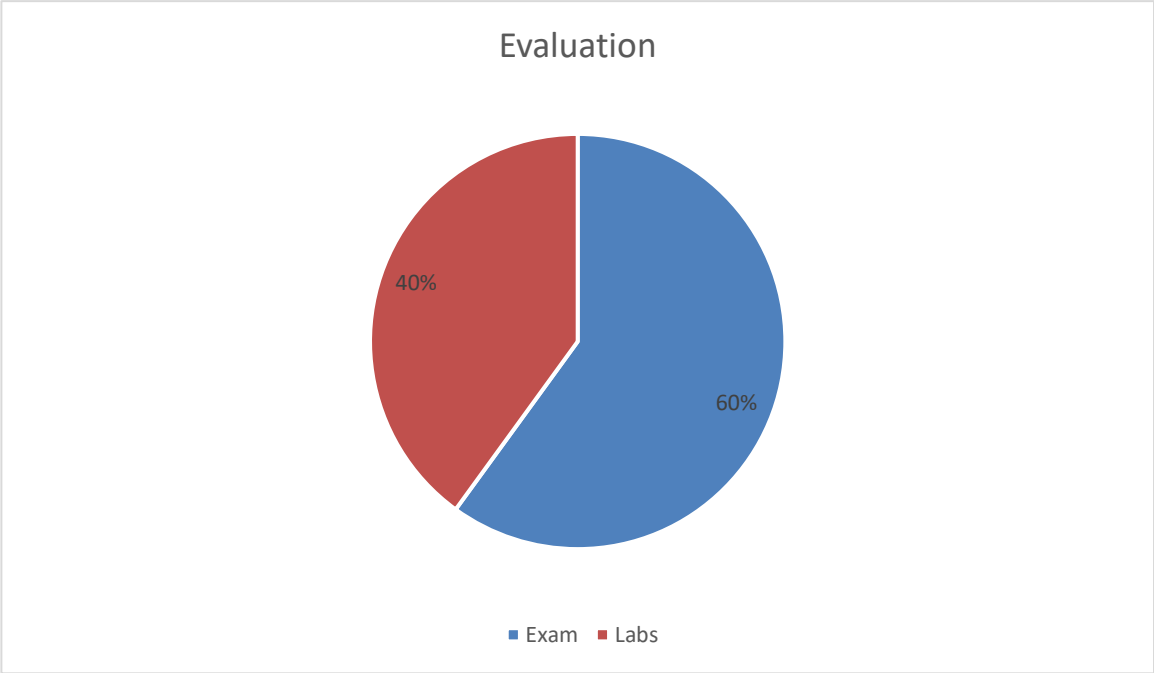
6. Predict and assess the outputs of any simple Python code.

6. Learning assessment methods

The final evaluation is done through:

- ***A final exam*** (marked out of 20) on a table which covers everything you have seen in this course during the semester.
- ***A Lab scores*** out of 20 (evaluation of the individual practical works submitted during the classes).

Note: The final exam is 60% from the final grade, and the lab is 40%



7. *Teaching-learning activities*

The course offers several methods with their specificity and advantages.

In person:

❖ Knowledge is disseminated through a lecture. Students are invited to take the fundamental notes to master the essential elements as best as possible and quickly. carrying out the learning activities proposed during the session.

❖ Students are also called to participate in debates and discussions initiated by the questions asked about the current educational sequence. No form of evaluation is performed. The objective is to promote exchanges between students and teachers to answer questions and compare points of view and draw educational benefits from these exchanges.

❖ Practical exercises are scheduled to train students to use all theoretical concepts presented, they also allow students to familiarize themselves with the algorithms and programming.

From a distance :

- ❖ Students are invited to interact via the Moodle platform offered by the university. They are called to :
- ❖ Participate in the "discussion" forum at the end of each chapter section by following the

thread discussion initiated by the teacher.

- ❖ face-to-face, this light version also contains learning activities inserted into the different educational sequences.
- ❖ Take the test at the end of each chapter, which should summarize what the student has learned and detect any gaps.
- ❖ Ask questions about the current chapter on the forum entitled “Ask your questions here”

8. *Educational alignment*

- ✓ The targeted skill is based on three pillars: knowledge; know-how and interpersonal skills. For essential knowledge, students acquire it through a lecture.
- ✓ For know-how, such as applying programming instructions to solve a problem, I will put the students in a situation of application through Lab exercises and problems to solve.
- ✓ For interpersonal skills (savoir-etre), this will allow students to make collective programming codes requiring the investment of all the concepts seen to write a Python program hat solve a problem, the students will work in groups, organize and present their ideas.

9. *Operating methods*

The course is structured in:

- Theoretical training aimed at providing you with the basic concepts essential to rapid identification of definitions, working methods to follow and tools necessary to the computer science.
- Practical sessions intended to promote the knowledge obtained during the course. The course takes place as a combination of face-to-face (in-class) and on-site lessons. remotely via the distance learning platform, which will allow you, if necessary, to review or to deepen the concepts seen in class and to overcome the difficulties encountered. The platform online contains spaces for:
- Download the different chapters.
- Submit your individual project in the appropriate space.

10. *Help Resources*

Here are some suggested references and resources that could be used for this introductory programming course:

Textbooks:

1. Introduction to Computing and Programming in Python: A Multimedia Approach by Mark Guzdial and Barbara Ericson Starting Out with Python by Tony Gaddis
2. Python Crash Course by Eric Matthes

Online References:

1. Python Documentation - Official Python language documentation: https://docs.python.org/3/
2. W3Schools Python Tutorial - Interactive tutorials covering Python syntax: https://www.w3schools.in/python-tutorial/ TutorialsPoint Python Tutorial - Tutorial covering Python basics: https://www.tutorialspoint.com/python/index.htm

Code Examples:

1. Rosetta Code - Collection of programming tasks with solutions in many languages: https://rosettacode.org/wiki/Category:Python
2. GitHub - Sample Python projects, code snippets and gists: https://github.com/topics/python