INTRODUCTION TO PROGRAMMING

Grado en Administración de Empresas / Bachelor in Business Administration BBA SEP-2023 PRO-NBA.1.M.C

Area Information Systems and Technology

Number of sessions: 20

Academic year: 23-24

Degree course: FIRST

Number of credits: 3.0

Semester: 2º

Category: COMPULSORY

Language: English

Professor: **ROGER CASTAÑER MÁRMOL**

E-mail: rcastaner@faculty.ie.edu

Roger Castañer Mármol has more than 20 years of experience across large and complex IT projects. Helping Customers make decisions to achieve their technical and business goals. Enthusiast with great customer-facing and teamwork skills. Leadership of national and international groups, in Traditional and Agile Methodologies with focus on results and cost optimization. Extensive experience in digital strategy and transformation programs.

Currently is Senior Manager for IT Academic area IE University & Business School

Roger got Degree CS at Universidad Alicante, Spain. Aside from work, in his spare time enjoys playing with his two young kids, teaching STEM to kindergarten and Primary School children, robotics and reading Sci-Fi books.

Office Hours

Office hours will be on request. Please contact at:

rcastaner@faculty.ie.edu
For urgent issues roger.castaner@ie.edu

Office Hours Wednesday 17.00 -18.00 Location: IE Tower

SUBJECT DESCRIPTION

Have you ever wondered how a computer works? Or how to develop the software your new start-up company needs?
?Computers are one of the most configurable machines we humans have invented since the dawn of time yet most of its users are constrained to uses designed by others (third party software). In this course you will learn how to code computer programs that will allow you to expand your computer's functionality up to its full potential.
?Knowing how to code is not something reserved for engineers or experts, everyone can write their own programs and if you combine your acquired business knowledge with the ability to write great software programs you could unleash a new set of opportunities for your career and boost your performance as an entrepreneur. If you are interested in knowing how to code software programs this course is for you.
?This is a course for beginners! Students joining this course do not need to have any previous coding experience as the course is designed for students learning how to code for the first time.

LEARNING OBJECTIVES

- Develop a logical way of thinking by carrying out programs
- Obtain skills to solve real problems through the use of programming languages.
- Achieve a strong foundation in programming with Python
- Introduction to problem solving for programming (i.e., "how to think about solving the problem" including techniques such as pseudo-code and specifications)
- Understanding basic object oriented programming and its importance in writing business software applications
- Being able to write fully functional console or GUI python programs
- Understanding programming concepts can help managers communicate more effectively with their technical teams. This can be especially useful when discussing project requirements and technical issues.
- Programming skills can help managers become more analytical in their decision making. Knowing the basics of Python helps better understand the built-in tools for data analysis and visualization, which can be used to inform business decisions. These aspects will be deepened in other courses of the program and this present course introduces the notions and concepts to adequately address them.
- Programming can be a useful problem-solving skill. Many business problems can be expressed in a way that is amenable to solution through programming, and being able to write code can help managers to identify and implement solutions more efficiently.
- Learning programming can be a valuable way for managers to broaden their knowledge and remain competitive in an increasingly technical business environment.
- As a manager, it is your responsibility to improve processes to add value to the customer or make the company more profitable. This will require new and improved software that automates bureaucracy and provides an improved way for the customer to interact with the company. The manager must be able to understand the potential of coding to make these improvements. The manager must also be able to communicate these needs to the developers so that they can translate them into code that meets the requirements.
- Programming requires that you first structure what you want to accomplish, determine how to get there and are able to express it in a very clear way. In other words, programming helps you think in an analytical way
- Efficient and effective algorithms
- Green IT.
- Responsible AI
- Sustainability impact measurement

TEACHING METHODOLOGY

The course is mainly a practical hands-on course. During the sessions we will mix theoretical lecturing with practical assignments. The course is designed for you to always have a laptop with you in class so you can code directly the exercises during the sessions. Approximately 50% of the time in the sessions we will have class discussions and theoretical lecturing together with small exercises to be completed during the session.

?Throughout the course there will be 4 practical sessions dedicated exclusively to reinforce the contents seen through more practical exercises.

?Additionally, you will have individual coding assignments that will help you reflect on what we have learned in class.

?Finally. you will develop a group project in order for you to familiarize with the concept of group programming as most of the real life software projects are implemented by a group of programmers so it is good for you to be familiar with organizing work and interacting with other coders.
?The last grade component will be your final exam during the last session.

IE University teaching method is defined by its collaborative, active, and applied nature. Students actively participate in the whole process to build their knowledge and sharpen their skills. Professor’s main role is to lead and guide students to achieve the learning objectives of the course. This is done by engaging in a diverse range of teaching techniques and different types of learning activities such as the following:

Learning Activity	**Weighting**	**Estimated time a student should dedicate to prepare for and participate in**
Lectures	21.33 %	16.0 hours
Discussions	10.67 %	8.0 hours
Exercises in class, Asynchronous sessions, Field Work	46.67 %	35.0 hours
Group work	10.67 %	8.0 hours
Individual studying	10.67 %	8.0 hours
TOTAL	100.0 %	75.0 hours

PROGRAM

SESSION 1 (LIVE IN-PERSON)

?Course introduction and overview of the computer programming environment?

Description: ?In this session we will learn about the history of computers and programming as well as review the course logistics and organization together with the computer programming environment (IDEs, compilers, etc…)?

Materials: ?Slides provided by the professor?

SESSION 2 (LIVE IN-PERSON)

?Data types & variables?

Description: ?In this session we will learn how to store information in our programs using variables and what are the different data-types for variables in Python. We will program our first “Hello World” program in the Python language.?

Materials: ?Slides provided by the professor

SESSION 3 (LIVE IN-PERSON)

?Writing programs?

Description: ?In this session, we will learn the process for planning and coding software?

Materials: ?Slides provided by the professor

SESSION 4 (LIVE IN-PERSON)

?Definite Loops and the Math and Random libraries?

Description: ?In this session we will introduce our first loops and the use of the Math and Random libraries.?

Materials: ?Slides provided by the professor

SESSION 5 (LIVE IN-PERSON)

Sustainability Topic: ?Throughout the previous sessions try to make students aware that smart programming contributes to sustainability by creating algorithms that optimize available resources.

?Practical session?

Description: ?In this session we will practice with more exercises based on the content covered in the previous sessions (1 to 4)?

Materials: ?Slides provided by the professor

SESSION 6 (LIVE IN-PERSON)

?Decision structures Part I?

Description: ?In this session we will expand the functionality we can code with iterations, logic and flow control with if/else structures.?

Materials: ?Slides provided by the professor

SESSION 7 (LIVE IN-PERSON)

Decision structures Part II??

Description: ?In this session we will expand the functionality we can code with iterations, logic and flow control with if/else structures. Exception Handling.?

Materials: ?Slides provided by the professor

SESSION 8 (LIVE IN-PERSON)

?Indefinite loops?

Description: ?In this session we will expand the functionality we can code with iterations, logic and flow control with indefinite loops.?

Materials: ?Slides provided by the professor

SESSION 9 (LIVE IN-PERSON)

Sustainability Topic: ?In the preceding sessions, discuss some examples of how decision structures contribute to improve sustainability through correct decision making.?

?Practical session?

Description: ?In this session we will practice with more exercises based on the content covered in the previous sessions (6 to 8)?

Materials: ?Slides provided by the professor

SESSION 10 (LIVE IN-PERSON)

Boolean Algebra?

Description: ?In this session we will expand the functionality we can code with the boolean datatype and Boolean algebra.?

Materials: ?Slides provided by the professor?

SESSION 11 (LIVE IN-PERSON)

?Functions, arguments and return types?

Description: ?In this session we will start encapsulating our code in methods that enable us to reuse the code preventing duplication and errors in large projects?

Materials: ?Slides provided by the professor?
Materials included in the syllabus

SESSION 12 (LIVE IN-PERSON)

?Lists and strings?

?In this session we will learn how to store multiple items in lists in Python and process text (strings).?

Materials: ?Slides provided by the professor?

SESSION 13 (LIVE IN-PERSON)

Sustainability Topic: ?Discuss in the previous sessions some examples of how programming with functions allows code reuse contributing to the optimization in the use of computing resources.?

Practical session?

Description: ?In this session we will practice with more exercises based on the content covered in the previous sessions (10 to 12)?

Materials: ?Slides provided by the professor?

SESSION 14 (LIVE IN-PERSON)

?Reading and writing files?

Description: ?In this session we will learn how to use files to store information on the long term memory.?

Materials: ?Slides provided by the professor?

SESSION 15 (LIVE IN-PERSON)

?Review: Lists, Functions and Files?

Description: ?In this session we will review some of the previous topics in order too consolidate knowledge.?

Materials: ?Slides provided by the professor

SESSION 16 (LIVE IN-PERSON)

?GUI programming ?

Description: ?In this session we will use the graphic library form PYTHON to develop our first simple GUI based program.?

Materials: ?Slides provided by the professor?

SESSION 17 (LIVE IN-PERSON)

?GUI programming ?

Description: ?In this session we will use the graphic library form PYTHON to develop our first simple GUI based program.?

Materials: ?Slides provided by the professor?

SESSION 18 (LIVE IN-PERSON)

Sustainability Topic: ?Now that students are able to create more sophisticated programs, the idea of how efficient programming contributes to important aspects of sustainability such as energy savings and smarter decision making supported by more accurate analytics will be emphasized.?

?Practical session ?

Description: ?In this session we will practice with more exercises based on the content covered in the previous sessions (14 to 17).?

Materials: ?Slides provided by the professor?

SESSION 19 (LIVE IN-PERSON)

?Practical session?

Description: ?This session will be dedicated to coding some practical exercises, which will involve making use of some of the most important topics of the course.?

Materials: ?Slides provided by the professor?

SESSION 20 (LIVE IN-PERSON)

Final exam?

Description: ?Final exam of the course.?

Materials: ?Guidelines provided in advance by the professor

EVALUATION CRITERIA

1. Class participation

You are expected to attend every class and participate in the discussions and class activities (games, etc.). The basic criteria in grading your participation are: a) your presence in each session, b) your (quality) contributions to the group discussion c) submission of activities. Lively discussions in the classroom are always encouraged, however, make sure that you provide constructive comments which contribute to the learning experience of the whole class.

2. Group assignment

A major part of this course's learning experience consists of a group project which will be focusing on designing and implementing an algorithmic solution to a problem. The definition of the problem as well as the main requirements will be provided to you during the sessions, so that you have only to focus on designing and implementing the best solution. Remember, creativity is always rewarded!

3. Final exam

At the end of the course you will have to pass an individual exam.

A minimum of 3.5 out of 10 on the final exam will be required to pass the course.This 20means that those students who obtain a grade lower than 3.5 in the final exam and their total final grade is higher than 5 out of 10, will still fail the course.

criteria	**percentage**	**Learning Objectives**	**Comments**
Final Exam	40 %		
Group Work	30 %		
Class Participation	10 %		
Intermediate tests	20 %		

RE-SIT / RE-TAKE POLICY

Each student has four (4) chances to pass any given course distributed over two (2) consecutive academic years. Each academic year consists of two calls: one (1) ordinary call (during the semester when the course is taking place); and one (1) extraordinary call (or "re-sit") in June/July.

Students who do not comply with the 80% attendance requirement in each subject during the semester will automatically fail both calls (ordinary and extraordinary) for that Academic Year and have to re-take the course (i.e., re-enroll) during the next Academic Year.

The Extraordinary Call Evaluation criteria will be subject to the following rules:

- Students failing the course in the ordinary call (during the semester) will have to re-sit evaluation for the course in June / July (except those students who do not comply with the attendance rule, and therefore will not have that opportunity, since they will fail both calls and must directly re-enroll in the course during the next Academic Year).
- It is not permitted to change the format nor the date of the extraordinary call exams or deadlines under any circumstance. All extraordinary call evaluation dates will be announced in advance and must be taken into consideration before planning the summer (e.g. internships, trips, holidays, etc.)
- The June/July re-sit will consist of a comprehensive evaluation of the course. Your final grade for the course will depend on the performance in this exam or evaluation only. I.e., continuous evaluation over the semester (e.g. participation, quizzes, projects and/or other grade components over the semester) will not be taken into consideration on the extraordinary call. Students will have to achieve the minimum passing grade of 5 and the maximum grade will be capped at 8.0 (out of 10.0) – i.e., "notable" in the extraordinary call.
- Re-takers: Students who failed the subject on a previous Academic Year and are now re-enrolled as re-takers in a course will need to check the syllabus of the assigned professor, as

well as contact the professor individually, regarding the specific evaluation criteria for them as re-takers in the course during that semester (ordinary call of that Academic Year). The maximum grade that may be obtained as a retaker during the ordinary call (i.e., the 3rd call) is 10.0 (out of 10.0).

After exams and other assessments are graded by the professor (on either the ordinary or extraordinary call), students will have a possibility to attend a review session (whether it be a final exam, a final project, or the final overall grade in a given course). Please be available to attend the session in order to clarify any concerns you might have regarding your grade. Your professor will inform you about the time and place of the review session.

- Students failing more than 18 ECTS credits after the June/July re-sits will be asked to leave the Program. Please, make sure to prepare yourself well for the exams in order to pass your failed subjects.
- In case you decide to skip the opportunity to re-sit for an exam or evaluation during the June/July extraordinary call, you will need to enroll in that course again for the next Academic Year as a re-taker, and pay the corresponding tuition fees. As you know, students have a total of four (4) allowed calls to pass a given subject or course, in order to remain in the program.

BIBLIOGRAPHY

Recommended

- John Zelle. (2016). *Python Programming: An introduction to computer Science.* 3rd. Franklin, Beedle & Associates inc. ISBN 9781590282755 (Printed)

BEHAVIOR RULES

Please, check the University's Code of Conduct here. The Program Director may provide further indications.

ATTENDANCE POLICY

Please, check the University's Attendance Policy here. The Program Director may provide further indications.

ETHICAL POLICY

Please, check the University's Ethics Code here. The Program Director may provide further indications.