Last update: 30-07-2019

220005 - Fundamentals of Programming

Coordinating unit: 205 - ESEIAAT - Terrassa School of Industrial, Aerospace and Audiovisual Engineering

Teaching unit: 723 - CS - Department of Computer Science

Academic year: 2019

Degree: BACHELOR'S DEGREE IN AEROSPACE TECHNOLOGY ENGINEERING (Syllabus 2010). (Teaching unit Compulsory)
BACHELOR'S DEGREE IN AEROSPACE VEHICLE ENGINEERING (Syllabus 2010). (Teaching unit Compulsory)

ECTS credits: 6 Teaching languages: Catalan, Spanish

Teaching staff

Coordinator: MARTA GATIUS VILA
JOSEFINA LOPEZ HERRERA

Others: FATOS XHAFA XHAFA- MARTA GATIUS- JOSEFINA LOPEZ HERRERA

Degree competences to which the subject contributes

Specific:

1. A basic understanding of the use and programming of computers, operating systems, databases and computer programs with applications in engineering

Transversal:

2. EFFICIENT ORAL AND WRITTEN COMMUNICATION - Level 1. Planning oral communication, answering questions properly and writing straightforward texts that are spelt correctly and are grammatically coherent.

Last update: 30-07-2019

220005 - Fundamentals of Programming

Teaching methodology

- Face-to-face lecture sessions. Lectures are given using digital presentations.
The presentations will be made available to students on the virtual campus before classes begin to help them follow them.
The assessment will be based on mid-semester examinations (or an optional final examination for students who fail the first one).
- Face-to-face practical work sessions
During practical work sessions, students work individually or in small groups of 2-3 on problems and questions under the lecturer's supervision. A collection of problems will be made available on the virtual campus.
Systems for self-assessment (with assessment criteria or rubrics), co-assessment (among students) and delivery of reports, corrected by the teacher and returned, are made available to facilitate independent learning.
- Laboratory work sessions Students work in pairs during laboratory sessions.
Guidelines for practicals will be available to students on the virtual campus at the start of the course. Students must hand in a report for each practical. Marks will be based on the work carried out in the laboratory and the reports handed in.

In the theory sessions the teachers will introduce new concepts
In the laboratory sessions the students will practice the new concepts
using computers. There will be two different types of laboratory sessions:
Sessions in which the teacher guide students in analyzing data and
solving problems by applying new concepts and techniques .
Exam sessions in which the students , using online tools, solve exams
that will be evaluated
Students will also have to work by them-selves in order

Learning objectives of the subject

The main goal of the subject Computer Science is to learn to develop programs using a high level language.In order to pass the subject the student has to be able to:
- Learn the basi concept related to software and hardware: computers structure and operative systems.
- Learn the basic concepts for programming.
- Develop the ability to use basic tools and basic techniques of programming:algorithms and programs.
- Learn to design programs correctly: well-structured, efficient and readable.
- Learn to design data structures representing the data involved in agiven problem.
- Design and develop a medium size program.
- Develop their ability to use abstraction in programming patterns to solve real problems.

Study load

Total learning time: 150h	Hours large group:	0h	0.00%
	Hours small group:	60h	40.00%
	Self study:	90h	60.00%

Last update: 30-07-2019

220005 - Fundamentals of Programming

Content

TOPIC 1: INTRODUCTION TO COMPUTERS

Learning time: 11h

Laboratory classes: 2h
Guided activities: 0h
Self study : 9h

Description:

1.1. Computer architecture.
1.2. Operating systems.
1.3. Computer programming.
1.4. Algorithms and programs.

Related activities:

Activity 5, which corresponds to a group research and development project on an introductory topic.

Specific objectives:

- Define the terms "hardware" and "software".
- Understand the general structure of a computer.
- Understand what an operating system is.
- Name and describe the various types of operating systems.
- Know what a program is.
- Name various programming languages.
- Name and describe the various programming paradigms.
- Understand the basic requirements of a program.
- Name and describe the phases of software development.
- Understand what an algorithm is.

Last update: 30-07-2019

220005 - Fundamentals of Programming

TOPIC 2: BASICS OF STRUCTURED PROGRAMMING	Learning time: 28h Laboratory classes: 10h Self study : 18h

Description:
2.1. Structure of a program.
2.2. Objects.
2.3. Expressions and operators.
2.4. Elementary actions.
2.5. Writing instructions.

Related activities:
- One type-1 activity: Individual continuous-assessment test spanning the various laboratory-group sessions.
- One or more type-2 activities: Individual take-home self-directed learning test.
- One or more type-3 activities: Individual take-home self-directed learning task.

Specific objectives:
- Properly use the objects in a program: constants and variables.
- Distinguish between correct and incorrect identifiers.
- Describe and use correctly the type of data available in the programming language as well as the defined operations.
- Describe the function of basic input and output actions and use them correctly.
- Understand assignment statements and use them correctly.
- Describe the structure of a program.
- Correctly develop a test suite.
- Determine whether the block structure of a program is correct.
- Correctly use alternative and iterative composition.
- Correctly build programs with objects, expressions, elementary actions and compositions.

Last update: 30-07-2019

220005 - Fundamentals of Programming

TOPIC 3: SUBROUTINES: ACTIONS AND FUNCTIONS	Learning time: 27h Laboratory classes: 10h Guided activities: 0h Self study : 17h

Description:
3.1. Actions.
3.2. Functions.
3.3. Parameter passing.
3.4. Library functions.

Related activities:
- One type-1 activity: Individual continuous-assessment test spanning the various laboratory-group sessions.
- One or more type-2 activities: Individual take-home self-directed learning test.
- One or more type-3 activities: Individual take-home self-directed learning task.
- One phase of activity 4, the project.

Specific objectives:
Determine whether a subroutine needs to be an action or a function.
- Send parameters by value and by reference.
- Define formal and actual parameters (arguments).
- Determine whether a formal parameter of an action or function is input, output or input/output.
- Write a program that uses library functions properly.
- Implement and use functions and actions properly.
- Detect and eliminate code repetition.
- Build programs correctly with the help of functions and actions.

220005 - Fundamentals of Programming

TOPIC 4: BASIC ALGORITHMIC TECHNIQUES

Learning time: 26h

Laboratory classes: 10h
Self study : 16h

Description:
4.1. Sequences.
4.2. Traversal techniques.
4.3. Search techniques.

Related activities:
- One type-1 activity: Individual continuousassessment test spanning the various laboratory-group sessions.
- One or more type-2 activities: Individual take-home self-directed learning test.
- One or more type-3 activities: Individual take-home self-directed learning task.
- One phase of activity 4, the project.

Specific objectives:
- Describe the concept of a sequence.
- Define the sequences associated with a problem.
- Given a sequence problem, determine whether the scheme can be solved by a search technique or a traversal technique.
- Correctly apply traversal and search algorithms.

4. Structured types

Learning time: 31h

Laboratory classes: 14h
Guided activities: 0h
Self study : 17h

Description:
5.1. Tuples.
5.2. Tables.
5.3. Traversal and search techniques in tables.
5.4. Sorting and search algorithms.

Related activities:
- One type-1 activity: Individual continuous-assessment test spanning the various laboratory-group sessions.
- One or more type-2 activities: Individual take-home self-directed learning test.
- One or more type-3 activities: Individual take-home self-directed learning task.
- One phase of activity 4, the project.

Specific objectives:
- Write the declaration of a tuple and a table.
- Correctly declare tuple and table variables and access the declarations properly.
- Correctly use pass-by-value and pass-by-reference evaluation with tuples and tables.
- Generate partially filled tables.
- Write code that inserts and removes items in a variable-length table.
- Understand and correctly use some sorting algorithms.

Last update: 30-07-2019

220005 - Fundamentals of Programming

5. Descendant design	Learning time: 27h Laboratory classes: 14h Guided activities: 0h Self study : 13h

Description:
6.1. Top-down design of data structures.
6.2. Top-down design of processes.

Related activities:
- Completion of activity 4, the project.

Specific objectives:
- Design an efficient data structure for a given problem.
- Correctly access complex data structures.
- Given a complex problem, carry out top-down design using subroutines.

Last update: 30-07-2019

220005 - Fundamentals of Programming

Planning of activities

ACTIVITY 3: LABORATORY EXAMS	Hours: 38h Laboratory classes: 18h Self study: 20h

Description:
Independent work in the classroom to cover all the specific learning objectives of the subject. Professors corrections.

Support materials:
Topic notes available (PowerPoint) available in ATENEA.
Activity statement and the official correction criteria (rubric) available in ATENEA.

Descriptions of the assignments due and their relation to the assessment:
Resolution of the exercise by the student.
The activities of type 1 represent 20% of the laboratory assessment.

Specific objectives:
At the end of the activity, the student must have achieved specific objectives of the subject.

ACTIVITY 2: INDIVIDUAL TESTS (CONTINUOUS ASSESSMENT IN ATENEA)	Hours: 2h Self study: 2h

ACTIVITY 3: INDIVIDUAL TASKS (CONTINUOUS ASSESSMENT IN ATENEA)	Hours: 3h Self study: 3h

ACTIVITY 4: MIDTERM EXAM	Hours: 43h Laboratory classes: 18h Self study: 25h

ACTIVITY 5: FINAL EXAM	Hours: 45h Laboratory classes: 20h Self study: 25h

ACTIVITY 6: PROJECT	Hours: 19h Laboratory classes: 4h Self study: 15h

Last update: 30-07-2019

220005 - Fundamentals of Programming

Qualification system

- First examination: 20%
- Second examination: 30%
- Laboratory sessions: 20%
- Application/practicals: 10%
- Final Project: 20%

As part of the evaluation of the project it is included here the evaluation of the common skills

Regulations for carrying out activities

· Any of the control laboratory not performed, will be considered as non-marked
· In no case can any documentation or digital support be used in the partial or final test

Bibliography

Basic:

Gatius, Marta [et al.]. Programació pràctica en C++ [on line]. Barcelona: Edicions UPC, 2010 [Consultation: 03/05/2019]. Available on: <http://hdl.handle.net/2099.3/36843>. ISBN 9788498804010.

Complementary:

Marco, J.; Xhafa, F.; Vázquez, P. P. Fonaments d'informàtica: pràctiques de laboratori [on line]. 2a ed. Barcelona: Edicions UPC, 2008 [Consultation: 06/05/2019]. Available on: <http://hdl.handle.net/2099.3/36801>. ISBN 9788483019689.

Xhafa, Fatos [et al.]. Programación en C++ para ingenieros. 2006. Madrid: Thomson, 2006. ISBN 8497324854.

Kernighan, Brian W.; Ritchie, Dennis M. El lenguaje de programación C. 2a ed. México: Prentice-Hall Hispanoamericana, 1991. ISBN 9688802050.

Others resources:

Dissemination: http://cartesius.upc.es/adminmat/programacio/desenvolupa

Hyperlink

Divulgació de la programació

Resource

Audiovisual material

Pàgina web de l'assignatura a atenea

Resource

Pildoras de C++: http://www.minidosis.org/#/cursos/FI

Resource

Computer material

Jutge

Resource