The fundamental objective of the Information Technology course is learn to program in a high-level language, and implement programs to solve problems in science and technical areas.

To pass the course, students should be able to:

- Understand the basic concepts associated with computer hardware and software: the structure of computers and
operating systems.

- Understand the fundamental concepts of computer programming.
- Develop skills in using basic tools and techniques of programming: algorithms and programs.
- Develop the capacity of abstraction in the use of programming schemes to solve real problems.
- Design well-structured and readable programs.
- Conduct a programming project of average complexity.

### Study load

<table>
<thead>
<tr>
<th>Total learning time: 150h</th>
<th>Hours large group: 30h</th>
<th>20.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours medium group: 0h</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>Hours small group: 30h</td>
<td>20.00%</td>
<td></td>
</tr>
<tr>
<td>Guided activities: 0h</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>Self study: 90h</td>
<td>60.00%</td>
<td></td>
</tr>
</tbody>
</table>
# Content

## 1. Introduction to Programming

<table>
<thead>
<tr>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1. Basics</td>
</tr>
<tr>
<td>1.2. Problem Solving Cycle with the Computer</td>
</tr>
<tr>
<td>1.3. Functional Structure of the Computer</td>
</tr>
<tr>
<td>1.4. Primitive Instructions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Related activities:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity 1: Problems about primitive instructions</td>
</tr>
<tr>
<td>Activity 4: Cuestionarios</td>
</tr>
<tr>
<td>Activity 5: Control 1</td>
</tr>
<tr>
<td>Activity 7: Control 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Learning time:</th>
</tr>
</thead>
<tbody>
<tr>
<td>10h</td>
</tr>
<tr>
<td>Theory classes: 2h</td>
</tr>
<tr>
<td>Laboratory classes: 2h</td>
</tr>
<tr>
<td>Self study : 6h</td>
</tr>
</tbody>
</table>

## 2. Data Types, Operations and Variables

<table>
<thead>
<tr>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1. Variables</td>
</tr>
<tr>
<td>2.2. Data Types</td>
</tr>
<tr>
<td>2.3. Expressions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Related activities:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity 1: Problems about writing and evaluating expressions</td>
</tr>
<tr>
<td>Activity 4: Questionnaires</td>
</tr>
<tr>
<td>Activity 5: Control 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Learning time:</th>
</tr>
</thead>
<tbody>
<tr>
<td>25h 15m</td>
</tr>
<tr>
<td>Theory classes: 4h</td>
</tr>
<tr>
<td>Laboratory classes: 5h 15m</td>
</tr>
<tr>
<td>Self study : 16h</td>
</tr>
</tbody>
</table>
### 3. Control Statements

**Learning time:** 28h 15m
- Theory classes: 4h
- Laboratory classes: 5h 15m
- Guided activities: 1h
- Self study: 18h

**Description:**
3.1. Sequential Construction
3.2. Conditional Construction
3.3. Iterative Construction
3.4. Functions

**Related activities:**
- Activity 4: Questionnaires
- Activity 5: Control 1
- Activity 7: Control 2

### 4. Sequences and Iterative Schemes

**Learning time:** 25h 45m
- Theory classes: 6h
- Laboratory classes: 3h 45m
- Guided activities: 1h
- Self study: 15h

**Description:**
4.1. Sequences, Lists and Strings
   - 4.1.1. Lists
   - 4.1.2. Strings
4.2. Sequence Schemes
   - 4.2.1. Linear Scheme
   - 4.2.2. Search Scheme
4.3. Generic Sequences
4.4. Data Flows

**Related activities:**
- Activity 4: Control 1
- Activity 6: Making a stage of the activity 6 corresponding to practices
### 5. Lists

**Learning time:** 6h  
Theory classes: 6h

**Description:**  
See catalan content.

**Related activities:**  
Activity 4: Questionnaires  
Activity 6: Finishing the activity 6 corresponding to practices  
Activity 7: Control 2

### 6. Matrices

**Learning time:** 21h 45m  
Theory classes: 4h  
Laboratory classes: 1h 45m  
Guided activities: 2h  
Self study: 14h

**Description:**  
6.1. Matrices  
6.2. Matrix Schemes  
   6.2.1. Linear Scheme  
   6.2.2. Search Scheme

**Related activities:**  
Activity 4: Questionnaires  
Activity 6: Making a stage of the activity 6 corresponding to practices  
Activity 7: Control 2

### 7. Modular Design and Problem Solving

**Learning time:** 20h 45m  
Theory classes: 2h  
Laboratory classes: 1h 45m  
Self study: 17h

**Description:**  
7.1. Scope and Visibility of Identifiers  
7.2. Design and Use of Modules

**Related activities:**  
Activity 6: Finishing the activity 6 corresponding to practices
### Planning of activities

#### 1. PROBLEMS ABOUT PRIMITIVE INSTRUCTIONS

**Hours:** 0h 15m  
Laboratory classes: 0h 15m

**Description:**
Solve a problem of the topic in course, in the classroom. The problem will be chosen at random from a list proposed by teachers. Individual activity.

**Support materials:**
List of problems available at Atenea.

**Descriptions of the assignments due and their relation to the assessment:**
Solution of the problem given by the student.
The sum of activities 1, 2 and 3, all with the same weight, represents 15% of the final grade.

**Specific objectives:**
At the end of the activity, the student should be able to:
- Know primitive instructions of the programming language studied.
- Run a simple program manually and show how input/output channels and memory have changed.

#### 3. PROBLEMS ABOUT EXPRESSION EVALUATION

**Hours:** 0h 15m  
Laboratory classes: 0h 15m

**Description:**
Solve a problem of the topic in course, in the classroom. The problem will be chosen at random from a list proposed by teachers. Individual activity.

**Support materials:**
List of problems available at Atenea.

**Descriptions of the assignments due and their relation to the assessment:**
Solution of the problem given by the student.
The sum of activities 1, 2 and 3, all with the same weight, represents 15% of the final grade.

**Specific objectives:**
At the end of the activity, the student should be able to:
- Know control statements in the programming language studied.
- Trace a program that contains control statements and understand what it does.
- Know when to use a control statement.
- Build programs by using correct variables, expressions and control statements.

#### 3. PROBLEMS ABOUT LISTS AND ITERATIVE SCHEMES

**Hours:** 0h 15m  
Laboratory classes: 0h 15m

**Description:**
Solve a problem of the topic in course, in the classroom. The problem will be chosen at random from a list proposed by teachers. Individual activity.

**Support materials:**
List of problems available at Atenea.
### Descriptions of the assignments due and their relation to the assessment:
- Solution of the problem given by the student.
- The sum of activities 1, 2 and 3, all with the same weight, represents 15% of the final grade.

### Specific objectives:
- At the end of the activity, the student should be able to:
  - Knowing the implementation of the sequences in the programming language studied.
  - Create, access and modify a sequence implemented as a list or as a string.
  - Apply iterative schemes on sequences.
  - Write correct programs that use iterative schemes on any sequence.

### 4. QÜESTIONARIS

**Description:**
Completion of self-assessed questionnaires (Atenea) to check up the subject comprehension.

**Support materials:**
- Self-assessed questionnaires available at Atenea.
- Collection of problem statements.
- Laboratory manual available at Atenea.

**Descriptions of the assignments due and their relation to the assessment:**
Students may complete each questionnaire along a deadline. All questionnaires are online. The whole set of questionnaires contributes 5% in the final grade.

### 5. CONTROL 1

**Description:**
Individual test which includes the first three issues of the course, consisting of solving a number of problems by hand.

**Support materials:**
- Exam questions.

**Descriptions of the assignments due and their relation to the assessment:**
- Solutions of the exam questions given by the student.
- Control 1 represents 25% of the final grade.

**Specific objectives:**
- At the end of the activity, the student must have achieved the specific objectives of topics covered by the exam.

### 6. PRACTICES

**Hours:** 36h
- Laboratory classes: 6h
- Self study: 30h
7. CONTROL 2

**Description:**
Individual test which includes all of the course topics, consisting of solving a number of problems by hand. The test will include some problem related with the practices.

**Support materials:**
Exam questions.

**Descriptions of the assignments due and their relation to the assessment:**
Solutions of the exam questions given by the student.
Control 2 represents 35% of the final grade.

**Specific objectives:**
At the end of the activity, the student must have achieved the specific objectives of topics covered by the exam.

---

**Qualification system**

\[ \text{Final Grade} = \max(50\% \ C_2, \ 20\% \ C_1 + 30\% \ C_2) + 5\% \ QU + 15\% \ AC + 30\% \ PR \]
Regulations for carrying out activities

Activities 1, 2, and 3, are individual and classroom activities.

Activity 4 is a series of self-assessed questionnaires that the student solves online and individually.

Activity 6 consists of several programs that solve problems of increasing difficulty must be delivered. These problems are solved in group. For each problem there is scheduled an online delivery. The teacher can ask students for an explanation of the work presented, and take into account their response to qualify the work.

Bibliography

Basic:


Others resources:

Hyperlink

http://marmota.act.uji.es/mtp/pdf/python.pdf

eBook that introduces programming in Python.

http://atenea.upc.edu/moodle/

Notes, collection of problem statements and lab manual for the Informatics course.