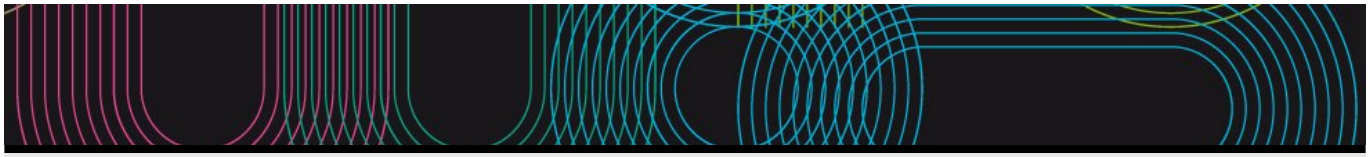

| IDENTIFYING DATA | | | | |
|---|---|---|---|---|
| **Programming I** | | | | |
| Subject | Programming I | | | |
| Code | V05G306V01105 | | | |
| Study programme | Grado en Ingeniería de Tecnologías de Telecomunicación | | | |
| Descriptors | ECTS Credits | Choose | Year | Quadmester |
| | 6 | Mandatory | 1st | 1st |
| Teaching language | English | | | |
| Department | | | | |
| Coordinator | Rodríguez Hernández, Pedro Salvador | | | |
| Lecturers | Costa Montenegro, Enrique<br>Rodríguez Hernández, Pedro Salvador | | | |
| E-mail | pedro.rodriguez@uvigo.es | | | |
| Web | http://moovi.uvigo.gal | | | |
| General description | The aim of the course is to provide students with basic skills to program in a high level language.<br><br>The programming paradigm followed is that of "structured programming".<br><br>English Friendly subject: International students may request from the teachers: a) materials and bibliographic references in English, b) tutoring sessions in English, c) exams and assessments in English. | | | |

| Training and Learning Results | |
|---|---|
| Code | |
| B4 | CG4: The ability to solve problems with initiative, to make creative decisions and to communicate and transmit knowledge and skills, understanding the ethical and professional responsibility of the Technical Telecommunication Engineer activity. |
| B9 | CG9: The ability to work in multidisciplinary groups in a Multilanguage environment and to communicate, in writing and orally, knowledge, procedures, results and ideas related with Telecommunications and Electronics. |
| C6 | CE6/T1: The ability to learn independently new knowledge and appropriate techniques for the conception, development and exploitation of telecommunication systems and services |
| C12 | CE12/T7: The knowledge and use of basics in telecommunication networks, systems and service programming. |
| D2 | CT2 Understanding Engineering within a framework of sustainable development. |
| D4 | CT4 Encourage cooperative work, and skills like communication, organization, planning and acceptance of responsibility in a multilingual and multidisciplinary work environment, which promotes education for equality, peace and respect for fundamental rights. |

| Expected results from this subject | |
|---|---|
| Expected results from this subject | Training and Learning Results |
| Express the solution of a simple problem by means of algorithms using top-down design. | C12 |
| Identify the data needed to solve a problem and associate them with appropriate datatypes based on their features (size, range, associated operators) | C12 |
| Code simple algorithms using the basic types of statements: assignment, selection and iteration. | C12 |
| Declare and define functions with a proper use of parameters. | C12 |
| Handle I/O operations and file management. | C12 |
| Define and use structured data types. | C12 |
| Define and manage dynamic data structures (lists, stacks, queues and trees). | C12 |
| Create modules and library functions and use them in programs. | C6<br>C12 |
| Predict the result of a sequence of statements, knowing the input data. | C12 |
| Handle basic tools in an integrated development environment: text editor, compiler, linker, debugger and documentation tools. | C6 |

| | | | |
|---|---|---|---|
| Develop a small scale project following all the phases: requirements analysis, design, implementation, testing and documentation. | B4<br>B9 | C6<br>C12 | D2<br>D4 |

**Contents**

| Topic | |
|---|---|
| Lecture 1: The algorithm and the programming languages. | 1. A computer's structure and operation<br>2. How the program gets into the computer<br>3. C Programming language<br>4. The process of developing programs<br>5. Simple Programming Examples<br>6. Software engineering concepts |
| Lecture 2: Grammar and basic elements of C language. | 1. Basic elements of a C program<br>2. Identifiers<br>3. Expressions<br>4. Declaration and initialization<br>5. The assignment statement<br>6. Formatted input/output |
| Lecture 3: Iteration and selection statements | 1. Control statements<br>2. Decision statements: (a) if statement (b) if-else statement (c) switch statement<br>3. Iteration statements: (a) do-while statement (b) while statement (c) for statement<br>4. Statements for altering the control flow: break and continue statements |
| Lecture 4: Arrays and pointers | 1. Data Structures<br>2. Arrays: (a) One-dimensional arrays (b) Two-dimensional arrays<br>3. Strings<br>4. Copy of arrays |
| Lecture 5: Functions | 1. Function declaration and definition<br>2. Functions with no parameters<br>3. C inter function communication: local, global and static variables<br>4. Functions with parameters by value |
| Lecture 6. Pointers | 1. Pointers<br>2. Pointer arithmetic<br>3. Dynamic memory allocation<br>4. Arrays and pointers<br>5. Pointers to pointers<br>6. Functions with parameters by reference<br>7. Command line arguments |
| Lecture 7: Files | 1. Introduction: Types of files<br>2. Text files in C<br>3. Declaration<br>4. File opening and closing<br>5. File management<br>6. Operations on characters<br>7. Operations on strings<br>8. Formatted operations |
| Lecture 8: Structured type variables | 1. Introduction: Structured data types<br>2. struct type. Declaration<br>3. struct type. Operations<br>4. Pointers and struct type<br>5. struct as parameters<br>6. Creation of data types |
| Lecture 9: Lists | 1. Introduction: the need for dynamic data structures<br>2. Dynamic data structures<br>3. Linked lists (a) Types (b) Most common operations |

**Planning**

| | Class hours | Hours outside the classroom | Total hours |
|---|---|---|---|
| Introductory activities | 2 | 0 | 2 |
| Lecturing | 24 | 24 | 48 |
| Laboratory practical | 30 | 20 | 50 |
| Laboratory practice | 4 | 20 | 24 |
| Objective questions exam | 2 | 18 | 20 |
| Problem and/or exercise solving | 1 | 5 | 6 |

*The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.

| Methodologies | |
|---|---|
| | Description |
| Introductory activities | Introduction to theoretical and practical activities. |
| Lecturing | Professors present the main theoretical contents related to the subject<br><br>These sessions can include the development of works and programs by the students.<br><br>Through this methodology the competencies CE12 and CT2 are developed. |
| Laboratory practical | During the first part of the term the student codifies, compiles and documents simple programs guided by the instructor.<br><br>In this laboratory, the Ubuntu Linux operating system and the gcc compiler will be used.<br><br>Some of these activities can require the submission of a report in order to be evaluated.<br><br>Through this methodology the competencies CG4, CG9, CE6, CE12, CT2, and CT4 are developed. |

| Personalized assistance | |
|---|---|
| **Methodologies** | **Description** |
| Lecturing | The professors will provide individual attention to the students along the term, solving their doubts and questions. Questions will be answered during the master sessions or during tutorial sessions. The professors will establish timetables for this purpose at the beginning of the term. This schedule will be published on the subject website. Tutorial sessions could also be agreed by appointment. The tutorial schedule of the professors is avaliable in their Moovi profiles: https://moovi.uvigo.gal/user/view.php?id=11584 https://moovi.uvigo.gal/user/view.php?id=11583 |
| Laboratory practical | The professors will provide individual attention to the students along the term, solving their doubts and questions about the laboratory practises. Questions will be answered during the lab sessions or during tutorial sessions. The professors will establish timetables for this purpose at the beginning of the term. This schedule will be published on the subject website. Tutorial sessions could also be agreed by appointment. The tutorial schedule of the professors is avaliable in their Moovi profiles: https://moovi.uvigo.gal/user/view.php?id=11584 https://moovi.uvigo.gal/user/view.php?id=11583 https://moovi.uvigo.gal/user/view.php?id=59589 https://moovi.uvigo.gal/user/view.php?id=11342 https://moovi.uvigo.gal/user/view.php?id=11665 https://moovi.uvigo.gal/user/view.php?id=11299 https://moovi.uvigo.gal/user/view.php?id=11585 https://moovi.uvigo.gal/user/view.php?id=11338 |

| Assessment | | | | | |
|---|---|---|---|---|---|
| | Description | Qualification | Training and Learning Results | | |
| Laboratory practice | The student will take 2 midterm laboratory tests consisting in the development of small programs on the computer.<br><br>Each of these tests will assess the student's progress on a portion of the laboratory practical exercises.<br><br>The final laboratory test will assess the student's progress on the practical exercises as a whole. | 50 | B4<br>B9 | C6<br>C12 | D2<br>D4 |
| Objective questions exam | The student will take 1 midterm theoretical test that may consist of:<br>- short answer questions<br>- multiple choice questions<br><br>This exam will assess individually the student's mastery of the concepts introduced in the lecture sessions.<br><br>The final theoretical exam will also contain this type of questions. | 40 | B4 | C12 | |
| Problem and/or exercise solving | The theoretical exams will have a part consisting of problem and/or exercise solving | 10 | B4 | C12 | |

| Other comments on the Evaluation |
|---|

Following the guidelines specific to the degree program, each student will have 2 opportunities (the **ordinary** and **extraordinary** calls) to pass the course.

Furthermore, in the ordinary call, there will be 2 evaluation procedures (**continuous** and **global**).

**ASSESSMENT TESTS**

Throughout the semester, several intermediate assessment tests will be given. Specifically, there will be two **Midterm LaboratoryTests** (PL1 and PL3) and one **Midterm TheoreticalTest** (PT2). The schedule of the different intermediate assessment tests will be approved by the Academic Degree Committee (CAG) and will be available at the beginning of the semester.

During the regular examination period of the School, the **Final Theoretical Test** (ETF) and the **Final Laboratory Test** (EFL) will be given.

During the extraordinary examination period of the School, the **Extraordinary Theoretical Test** (ETX) and the **Extraordinary Laboratory Test** (EXL) will be given.

Each theoretical test may include short-answer and/or multiple-choice questions, as well as problem-solving and/or exercise resolution questions. It assesses students' knowledge of the content covered in the lectures.

All the practical exercises are mandatory. Prior to each laboratory test, it will be necessary to have uploaded to Moovi all the corresponding assignments for that test. Each laboratory test consists of making modifications to the submitted practical exercises. It evaluates those submitted practical exercises.

---

**ORDINARY CALL**

Each student taking this course may choose between the 2 evaluation procedures: continuous assessment and global assessment.

Taking the second midterm test (PT2) will be interpreted as the decision to choose continuous assessment. No taking it will be interpreted as the decision to choose global assessment.

---

**CONTINUOUS ASSESSMENT**

The condition for passing the course using the continuous assessment procedure is obtaining a final grade (NFC) equal to or higher than 5.

The final grade for continuous assessment will be calculated as the weighted arithmetic mean of the midterm and final test grades. It will be given by the following expression:

NFC = 0.6 NPP + 0.2 ETF + 0.2 EFL

Where:

NPP is the Midterm Test Grade, calculated as the weighted arithmetic mean of all midterm tests, according to the following expression:

NPP = (1PL1 + 3PT2 + 2*PL3) / 6

ETF is the grade obtained on the Final Theoretical Test.

EFL is the grade obtained on the Final Laboratory Test.

A minimum grade of 2.5 points will be required in the three components of this grade (NPP, ETF, and EFL). If the student fails to reach this minimum in any of them, the final grade for continuous assessment will be at most 4.0 (Fail).

Continuous assessment consists of the tests detailed in this guide, which are not recoverable. In other words, if a student cannot complete them within the specified timeframe, the teaching staff is not obliged to repeat them.

Before each test, the date and procedure for reviewing the grades will be indicated. Students will have the option to know the grade of each test and review the correction within approximately 2 weeks.

---

**GLOBAL ASSESSMENT**

The condition for passing the course using the global assessment procedure is obtaining a final grade (NFG) equal to or higher than 5.

This method will consist of the same final tests as the continuous assessment, although with different weights. The final grade for global assessment will be given by the following expression:

NFG = (ETF + EFL) / 2

A minimum grade of 2.5 points will be required in the two components of this grade (ETF and EFL). If the student fails to reach this minimum in any of them, the final grade for global assessment will be at most 4.0 (Fail).

---

Each student taking the final tests for the course will have both grades calculated: the final grade for continuous assessment (NFC) and the final grade for global assessment (NFG). The higher of the two grades will be awarded as the final grade in the ordinary call.

---

The grade will be "No-show" if the student does not attend any test after the first Midterm Test (PL1).

---

**EXTRAORDINARY CALL**

Each student who does not pass the course in the ordinary call will have a second opportunity.

In the extraordinary call, the condition for passing the course is obtaining a final grade (NFX) equal to or higher than 5.

The final grade in the extraordinary call will be given by the following expression:

NFX = (NTX + NXL) / 2

Where:

NTX is the Extraordinary Theoretical Grade: if the student takes the Extraordinary Theoretical Test, NTX will be the grade obtained in that test:

NTX = ETX

If not, NTX will be the theoretical grade obtained in the ordinary call:

NTX = 0.6 PT2 + 0.4 ETF

NXL is the Extraordinary Laboratory Grade: if the student takes the extraordinary Laboratory Test, NXL will be the grade obtained in that test:

NXL = EXL

If not, NXL will be the laboratory grade obtained in the ordinary call:

NXL = 0.2 PL1 + 0.4 PL2 + 0.4 EFL

A minimum grade of 2.5 points will be required in the two components of this grade (NTX and NXL). If the student fails to reach this minimum in any of them, the final grade in the extraordinary call will be at most 4.0 (Fail).

---

**END-OF-PROGRAM TEST**

Following the guidelines specific to the degree program, students who have 3 or fewer courses remaining to graduate will have end-of-program test call in those courses.

In the end-of-program test call, the condition for passing the course is obtaining a final grade (NFX) equal to or higher than 5.

In this call, a test with short-answer and/or multiple-choice questions, as well as problem-solving and/or exercise resolution questions, will be conducted (End-of-program Theoretical Test, ETZ), and a laboratory test evaluating the lab work (End-of-program Laboratory Test, ELZ). The final grade in the end-of-program test call will be given by the following expression:

NFZ = (ETZ + ELZ) / 2

A minimum grade of 2.5 points will be required in the two components of this grade (ETZ and ELZ). If the student fails to reach this minimum in any of them, the final grade in the end-of-program test call will be at most 4.0 (Fail).

---

The grade obtained in any of the assessable tasks will be valid only for the academic year in which they are performed, meaning that no grade is carried over from one year to the next.

---

If plagiarism is detected on any of the assignments/tests, the grade will be Fail (0), and the teaching staff will report the incident to the School's administration for appropriate action to be taken.

---

The use of generative artificial intelligence (GAI) is permitted in the completion of academic activities for this subject. Its use must be ethical, critical, and responsible. If GAI is used, any results it provides should be critically evaluated, and any generated citations or references must be carefully verified. Additionally, it is recommended to disclose the tools used.

---

## Sources of information

### Basic Bibliography


Brian W. Kernighan, Dennis M. Ritchie, **The C Programming Language**, 1995, Prentice Hall, 1983

Brian W. Kernighan, Dennis M. Ritchie, **El Lenguaje de Programación C**, 1995, Prentice Hall, 1983

Manuel Caeiro Rodríguez, Enrique Costa Montenegro, Ubaldo García Palomares, Cristina López Bravo, J, **Practicar Programación en C**, 2014,


### Complementary Bibliography


Ignacio Alvarado Aldea, Jose María Maestre Torreblanca, Carlos Vivas Venegas, Ascensión Zafra Cabeza, **100 Problemas Resueltos de Programación en Lenguaje C para Ingeniería**, 2017, Paraninfo, 2017

https://www.tutorialspoint.com/cprogramming/, **Learn C Programming**, 2021,

https://www.programiz.com/c-programming, **Learn C Programming**, 2021,

Stephen G. Kochan, **Programming in C**, 2014, Addison Wesley, 2005

Osvaldo Cairo Battistuti, **Fundamentos de Programación**, 2006, Pearson Education,

José Rafael García-Bermejo Giner, **Programación Estructurada en C**, 2008, Prentice Hall,

James L. Antonakos, Kenneth C. Mansfield Jr., **Programación Estructurada en C**, 2004, Prentice Hall, 1997

Jorge A. Villalobos S., Rubby Casallas G., **Fundamentos de Programación: Aprendizaje Activo Basado en Casos**, 2006, Prentice Hall,


## Recommendations

### Subjects that continue the syllabus

Informatics: Computer Architecture/V05G301V01109
Programming II/V05G301V01110

### Other comments

Programming II course continues this course in the second semester of the first year.