The study guide for the course

Universidad Autónoma de Madrid, Spain

1. Subject Identification

NAME	Introduction to Programming in C and Linux Operative System		CODE	
DEGREE	Computer Science	CENTER	EscuelaPolitécnica Superior	
TYPE	Compulsory	E.C.T.S.	3	
PERIOD	Semester	LANGUAGE	English	
COORDINATOR LECTURER		PNONE/EMAIL		ADDRESS
David Camacho		David.camacho@uam.es		
OTHER LECTURERS		PHONE/EMAIL		ADDRESS
Gema Bello Orgaz Antonio González Pardo Héctor Menéndez Benito		Gema.bello@uam.es Antonio.gonzalez@uam.es Hector.menendez@uam.es		

2. ContextDegree

Computer Sciences is widely extended in every society. The introduction of the Operative Systems has supposed an important improvement in every community gaining importance in fields such as health, education, administration, among others. In order to face these new technologies, it is important to understand and control them. In this context it is also important to understand the basis of software development starting with programming.

This course presents the basis of Computer Science degree which is essential during the whole degree such as programming and Linux advance management. In order to guarantee that the student acquires all the knowledge, the course has videoconference lectures, literature studies, tutorials and practical lectures which combine individual work and collaborative work.

It has 3 ECTS credits, which represent a total of 75 hours, 20 on-campus activities hours and 55self-study hours which also include collaborative work.

From the organizational point of view, the subject will have two hours of lectures a week (for a total of 20 hours) which includeslaboratory practice (for a total of 8 hours) and 2 group tutorials hours.

3. Requirements

Participants should have a sufficient knowledge of the English language that would allow them to follow English speaking classes, discuss, work collaboratively with students from other countries and write correctly in English.

• Participants should have experience working with Moodle.

• Participants should have some knowledge on instant messaging, social networking and social media.

• Participants have to be willing to work collaboratively.

4. Competenciesand Learning Outcomes

The computing-specific competences acquired by students through the IPCL course is: "Knowledge and application of the tools required to store, process and access information systems, including web-based systems".

Competencies

CG1	To characterize the computer science discipline.
CG2	To identify, describe and relate the tasks performed by an operating system. Use Linux-based programming environments.
CG3	To design, codify, debug, test, execute and interpret programs in the C procedural programming language.
CG4	To apply modularization as a specific technique for structuring programs.
CG5	To characterize the computer science discipline.

LearningOutcomes

The learning outcomes that students will achieve after completing the course are as follows:

LO1	• Knowledge of different operating systems at user (intermediate) and administrator (advanced) level.
LO2	• Ability to use effectively and proficiently programming environments, including

	code editing, compilation, assembly and debugging.	
LO3	• Ability to design and test a program to check for correctness, efficiency, usability and reliability.	
LO4	• Goodprogrammingpractices.	
LO5	• Use of some specialized computer applications.	
LO6	• Use of project management, release management, collaborative work tools, etc.	

5. Syllabus

TopicI.-   Introduction to computer science.

Topic II.- Basic concepts about operating systems.

Topic III.- Getting started with programming tools

Topic IV.- Advanced use of programming tools.

6. Working plan and methodology

Course schedule:

Content (topics)	Study week	Lectures	Seminars / Laboratory
1. **Introduction to Computer Science**	Week 1	Online lecture 18th September 11:00-13:00	Lecture (*Seminar-Documentation*)
2. **Basic Operative Systems**	Week 2	Online lecture 25th September 11:00-13:00	Lecture (*Seminar-Video*)
3. **Basic Programming.**	Week 3	Online lecture 9th October 11:00-13:00	Practical Task (*Laboratory-Foro-Chat*)
4. **IDE programming (S1)**	Week 4	Online lecture 16th October 11:00-13:00	Practical Task (*Laboratory-Foro-Chat*)
5. **IDE programming (S2)**	Week 5	Online lecture 23th October 11:00-13:00	Practical Task (*Laboratory-Online*)
6. **IDE programming (S3)**	Week 6	Online lecture 30th October 11:00-13:00	Practical Task (*Laboratory-Foro-Chat*)

	code editing, compilation, assembly and debugging.	
LO3	• Ability to design and test a program to check for correctness, efficiency, usability and reliability.	
LO4	• Goodprogrammingpractices.	
LO5	• Use of some specialized computer applications.	
LO6	• Use of project management, release management, collaborative work tools, etc.	

5. Syllabus

TopicI.-   Introduction to computer science.

Topic II.- Basic concepts about operating systems.

Topic III.- Getting started with programming tools

Topic IV.- Advanced use of programming tools.

6. Working plan and methodology

Course schedule:

Content (topics)	Study week	Lectures	Seminars / Laboratory
1. **Introduction to Computer Science**	Week 1	Online lecture 18th September 11:00-13:00	Lecture (*Seminar-Documentation*)
2. **Basic Operative Systems**	Week 2	Online lecture 25th September 11:00-13:00	Lecture (*Seminar-Video*)
3. **Basic Programming.**	Week 3	Online lecture 9th October 11:00-13:00	Practical Task (*Laboratory-Foro-Chat*)
4. **IDE programming (S1)**	Week 4	Online lecture 16th October 11:00-13:00	Practical Task (*Laboratory-Foro-Chat*)
5. **IDE programming (S2)**	Week 5	Online lecture 23th October 11:00-13:00	Practical Task (*Laboratory-Online*)
6. **IDE programming (S3)**	Week 6	Online lecture 30th October 11:00-13:00	Practical Task (*Laboratory-Foro-Chat*)

Content (topics)	Study week	Lectures	Seminars / Laboratory
7. IDE programming (S4)	Week 7	Online lecture 6th November 11:00-13:00	Practical Task (*Laboratory-Foro-Chat*)
8. Advance programming (S1)	Week 8	Online lecture 13th November 11:00-13:00	Practical Task (*Laboratory-Foro-Chat*)
9. Advance programming (S1)	Week 9	Online lecture 20th November 11:00-13:00	Practical Task (*Laboratory-Foro-Chat*)
10. Advance programming (S1)	Week 10	Online lecture 27th November 11:00-13:00	Practical Task (*Laboratory-Online*)
11. Advance programming (S1)	Week 11	Online lecture 4th December 11:00-13:00	Practical Task (*Laboratory-Foro-Chat*)
12. Advance programming (S1)	Week 12	Online lecture 11th December 11:00-13:00	Practical Task (*Laboratory-Foro-Chat*)
5.Assessment			
Practical Work, oral presentation			
Theoretical test	Week 4	Virtual Campus	
Theoretical test	Week 13	Virtual Campus	

Students will carry out face-to-face and self-study works and teachers will supervise these activities.

Teaching activities will be of one of these five types:

- Lectures, where to establish fundamental contents and where the student will be guide for their self-study activities.

- Workshops and seminars, to drive the student through active and collaborative learning, integrating lectures and virtual campus work.

- Practices, where to make different projects to solve any proposed problems. Individual and team projects will be done, requiring for students self-study work.

- Evaluation Sessions, examinations will be carried on in order to assess student acquisition of knowledge.

7. Learning material

Learning material consists of:

- scientific literature;

- additional interesting and useful literature;

- records of presentations and online consultations in virtual learning environment;

- practical tasks;
- real time chat, discussion forums and reflection blogs;
- video seminars and presentations.

Methodology

The student will have access to the following contents:

1. Learning Guide. It describes the basic theoretical elements that will be dealt with in the online lectures.
2. Learning Objects: Basically, the learning objects are the contents of the guide, along with some additional elements (videos, audio, etc.) and the self-assessment tools. These objects will be available in the Virtual Campus.
3. Additional books and readings: These are references to other additional learning contents that expand those of the Guide and the Learning Objects. The working method of this subject is described below:

   a) Before an online conference (minimum of 24 h.), the student will review the relevant content in the tutorial. It is of interest that will expand your knowledge with the materials of the Virtual Campus.
   b) During the conference on theoretical line, this content will again be reviewed. The content will also be expanded, using the annotated references.
   c) After the lecture, the student will have to revise this document or new learning objects, and additional references that are marked as required reading
   d) Required readings, will be scored as such in the Learning Guide. This review will be completed within 10 hours of video.
   e) The student must participate continuously in the forums virtual campus and in the development of the glossary.
   f) Practical activities will be defined for periods of 3 or 4 weeks.

NOTE: Students who do not pass the practical activities may request an evaluation through a comprehensive examination

8. LearningAssessment

The course in divided into four modules each with their own assessment method and criteria indicated. The first assessment represents 20% of the final grade. Assessment #2 and Assessment #3 represent 25% each of them. Finally, the last 30% of the grade corresponds to Assessment #4. These assessments could be quizzes and questionnaires or a programming assessment.

9. Resources, bibliography and complementary documentation

• J. G. Brookshear. Computer Science: An Overview. (9ª ed.)

- Banahan, M.; Brady, D.; Doran, M. (1991). The C Book (2nd ed.)
- King, K. N. (April 2008). C Programming: A Modern Approach (2nd ed.)
- Netbeans. C/C++ Application Learning Trail. http://netbeans.org/kb/trails/cnd.html
- Configuring the NetBeans IDE for C/C++. http://netbeans.org/community/releases/68/cpp-setup-instructions.html
- Eclipse for C/C++ Developers. http://www.eclipse.org/callisto/c-dev.php
- B. W. Kernighan, D. Ritchie, The C Programming Language. (2nd ed.)