We welcome you to study at the School of Computer Science at the University of Malaga (ETSI Informática – UMA). In this course guide, which is first and foremost designed to help exchange students to plan their studies during the exchange period, you will find descriptions of all the courses taught in English at our School during the academic year 2016-2017. Further inquiries will be handled by your contact teachers and the International Coordinator of the School:

Ana Cruz Martín  
subdir-iem@informatica.uma.es

Also the International Office is there to help you:  
Mª Carmen González  
mcgonzo@uma.es  
http://www.uma.es/relaciones-internacionales/

We hope you will enjoy your stay at University of Malaga.  
http://www.uma.es

More info: http://www.informatica.uma.es
CODE 201
NAME Analysis and Design of Algorithms
Credits 6 ECTS
Period Fall Semester

Course Specifications
In-classroom activities: lectures, problem solving sessions, laboratory work; Individual work: problem assignments.

Objectives and contents
The objectives of the course are (1) knowing and grasping the main techniques for algorithm design (i.e., being able to apply these for solving specific problems and being able to reason about their applicability and suitability) and (2) attaining adequate knowledge about algorithmic complexity (i.e., analyze the algorithms built, reason about their efficiency and perform comparisons among algorithms).

The course is organized in two major thematic parts, one dealing with algorithm analysis (complexity, formal specification and verification) and another one tackling algorithm design (comprising techniques such as divide-and-conquer, dynamic programming, greedy algorithms, backtracking and branch-and-bound).

Assessment
A continuous assessment methodology is used: four partial exams and several lab tests will be conducted during the semester. The final mark will be based on their outcome plus a bonus obtained by actively participating in class (e.g., delivering proposed assignments, solving problems in class, etc.). A final exam will be available for students who could not obtain enough points during the semester.

Lecturer
Dr. Carlos Cotta Porras
ccottap@lcc.uma.es
Room 3.2.49
CODE 202
NAME Databases
Credits 6 ECTS
Period Fall Semester

Course Specifications
Lectures where theoretical concepts are explained, and laboratory work where every student put in practice these concepts using a wide spread database system. Some problems have to be resolved in group. The course follows a practical approach.

Objectives and contents
The main objective is to learn the concept of relational database and the operations that can be made on it by using several tools. The second objective is to design correct and effective relational databases. Contents include:
- Database systems: an introduction
- Designing databases. The Entity/Relationship model
- Relational databases.
  - Definition of data structures and manipulation of data
  - Normalization
  - GUI tools for databases

Assessment
- Practical exam focused on SQL
- Practical exam focused on advanced SQL
- Report on a particular database problem and its solution by means of an E/R diagram and a relational database
- Participation in class

Lecturer
Dr. Sergio Gálvez Rojas
galvez@uma.es
Room 3.2.28

CODE 203
NAME Computer Organization
Credits 6 ECTS
Period Fall Semester

Course Specifications
Lectures in the classroom, individual work, questionnaires on-line by virtual campus and laboratory work

Objectives and contents
The objective of the subject is to understand the internal organization of a computer (from a hardware point of view). The contents are divided in three units. The first one is composed of two chapters: Measuring and understanding performance (~2 weeks) and enhancing performance with Pipelining (~4 weeks). This unit has a laboratory exercise: Simulation of a pipelined processor (DLX). The second unit is composed by an unique chapter: Memory hierarchy (~4 weeks) and we use a cache simulator in the laboratory. Finally, the last unit is composed by one chapter: Input/output and peripherals (~3 weeks) with a laboratory work: Example of an Input/output system: MIPS processor on a FPGA

Assessment
There will be four partial exams (~80% of the final grade) and other exercises (~20% of the final grade): list of exercises, problems, workshops, and practical assignments in the laboratory. A final exam is mandatory if the partial exams are failed.

Lecturer
Dr. Julio Villalba Moreno
jvillalba@uma.es
Room 2.2.38
Courses Syllabi

**CODE 204**

**NAME** Data Structures

**Credits** 6 ECTS

**Period** Fall Semester

**Course Specifications**

Lectures and laboratory work.

**Objectives and contents**

Data structures are nowadays used to organize huge amounts of data so that algorithms can process them efficiently. This course introduces most important data structures used in Computer Science. Different implementations of the structures using an object oriented programming language and a functional one are described and corresponding performances are analyzed and compared. Applications of data structures to solve different problems are also presented.

**Contents**

- Introduction to Functional Programming.
- More on Functional Programming.
- Linear data structures: stacks, queues and lists.
- Trees: search trees, heaps and balanced trees.
- Hash tables: hash function and, collision resolution techniques.
- Graphs: directed and undirected graphs, depth and breadth first traversals, topological sorting.

**Assessment**

Continuous assessment during the lectures and laboratory sessions. Written and on computer examinations.

**Lecturer**

Dr. José E. Gallardo  
pepeg@lcc.uma.es  
Room 3.0.5

---

**CODE 205**

**NAME** Automata Theory and Formal Languages

**Credits** 6 ECTS

**Period** Fall Semester

**Course Specifications**

The course is mainly based in lectures in which student participation is encouraged through the presentation of exercises, individual work, and group work. Extra activities include writing essays and solving proposed exercises.

**Objectives and contents**

The course gives a basic introduction to the classic and contemporary theory of formal languages and automata theory. Contents include the following topics: Mathematical preliminaries, Alphabets and Grammars, Regular Expressions, Finite automata, Regularity conditions, Context Free Languages, Introduction to computation, The Turing Machine, Recursive functions, The "While" language, Equivalence theorem, Universality, Formal limits of computation.

**Assessment**

The final grade will be obtained by adding up the partial grades obtained from the three types of proposed activities. A minimum grade of 5 should be obtained to pass the course, up to a maximum of 10 from the 11 points available. Activities: a) 3 partial tests during the course: Maximum grade: 4.5. b) Class participation, exercises presentation, essays, etc.: Maximum grade:1.5. c) Final exam: Maximum grade: 5.

**Lecturer**

Dr. Leonardo Franco  
lfranco@lcc.uma.es  
Room 3.2.29
<table>
<thead>
<tr>
<th>CODE</th>
<th>NAME</th>
<th>Credits</th>
<th>Period</th>
<th>Course Specifications</th>
<th>Objectives and contents</th>
<th>Assessment</th>
<th>Lecturer</th>
</tr>
</thead>
</table>
| 920   | Computational Intelligence          | 4.5 ECTS| Fall Semester  | The course is based on lectures in which the main ideas of the course are given as specific details of the problems and algorithms to be discussed and applied should be taken from reading the suggested related scientific publications. For each of the topics of the course, a computer based practice should be carried out together with a report of the work done and the results obtained. | The objective of the course is that the students acquire the knowledge and abilities needed in order to choose and apply computational intelligence algorithms for solving real problems in bioinformatics. | Course assessment is based on class participation, on the elaboration of reports and on a final exam.                                                                                                                                                                | Dr. Leonardo Franco  
  lfranco@lcc.uma.es  
  Room 3.0.29 |
### Course Specifications
Lectures, exercises and laboratory, where we work on several assignments in a team of maximum two persons. The educational material is based on a book, exercises and laboratory assignments.

### Objectives and contents
Go into the wonderful world of the computer processor. The student designs processors based on Boolean logic and observes its functioning based on code we feed it. We get a feeling for the language the processor understands based on creating assembler code.

### Assessment
The English of the student is promoted due to interacting in the lectures. The assessment of the course is done via a written and practical exam and hand in work of assignments during the course.

### Lecturer
Dr. Eligius M.T. Hendrix
[eligiushendrix@uma.es](mailto:eligiushendrix@uma.es)
https://sites.google.com/site/eligiushendrix/

---

<table>
<thead>
<tr>
<th>CODE</th>
<th>Name</th>
<th>Credits</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>109</td>
<td>Object Oriented Programming</td>
<td>6 ECTS</td>
<td>Spring Semester</td>
</tr>
<tr>
<td>110</td>
<td>Computer Technology</td>
<td>6 ECTS</td>
<td>Spring Semester</td>
</tr>
</tbody>
</table>

---

**Course Outline:**
1. **Object Oriented Programming**
   - Fundamentals: classes, objects, methods, messages, composition, inheritance, data polymorphism, dynamic binding.
2. Introduction to Java.
3. Exception management.
4. Basic predefined classes in java.lang, java.util and java.io.
5. Collections.
6. Graphical User Interfaces.

---

**Assessment**
Attending the lectures and the laboratory sessions, short questions to be answered during lectures, short programs to be developed during laboratory sessions, partial and final exams in the laboratory.

**Lecturers**
Dr. Mónica Pinto Alarcón
[pinto@lcc.uma.es](mailto:pinto@lcc.uma.es)
Room 3.2.7

Dr. Juan Miguel Ortiz de Lazcano
[jmortiz@lcc.uma.es](mailto:jmortiz@lcc.uma.es)
Room 3.2.21

---


<table>
<thead>
<tr>
<th>CODE</th>
<th>206</th>
<th>CODE</th>
<th>207</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>Introduction to Software Engineering</td>
<td>NAME</td>
<td>Systems Programming and Concurrency</td>
</tr>
<tr>
<td>Credits</td>
<td>6 ECTS</td>
<td>Credits</td>
<td>6 ECTS</td>
</tr>
<tr>
<td>Period</td>
<td>Spring Semester</td>
<td>Period</td>
<td>Spring Semester</td>
</tr>
</tbody>
</table>

### Course Specifications

**Introduction to Software Engineering**

The course is mainly based on lectures in which student participation is encouraged through classroom debates and collaborative development of exercises. Students will work on assignments both individually and in groups. Laboratory work will be used to teach selected software engineering tools and to develop the ability to apply the theoretical knowledge in practical scenarios. Extra activities include peer reviews, writing essays and solving proposed exercises.

**Objectives and contents**

This is an introductory course in Software Engineering. As such, and given the wide domain of the subject in terms of techniques and tools, the course is necessarily dense in content. We offer a broad view of the problems that are related to software development, the concepts, processes, techniques, tools and standards that conform what is called the Software Engineering Body of Knowledge. Our goal is to provide students with a good understanding of this central aspect of software development that will allow them to take more advanced courses on specific subfields.

**Assessment**

Continuous assessment of the student participation in lectures, debates and laboratory sessions throughout the course. Evaluation of student assignments. Written and practical (on computer) examinations.

**Lecturer**

Dr. Antonio Maña Gómez  
amg@lcc.uma.es  
Room 3.2.16

---

**Systems Programming and Concurrency**

Lectures where theoretical concepts, problems and algorithms are explained and laboratory work where every student put in practice these concepts resolving well-defined problems. The course follows a practical approach using the Java and C programming languages.

**Objectives and contents**

The main objective is to learn parallel programming in contrast to traditional programming models. Classical problems are shown as well as their solutions using several approaches. A second objective is to learn the C programming language from a low level point of view (system programming). Contents include:

2. Low level programming in C.
3. Concurrency: main concepts.
4. Concurrency: communication and synchronization.
5. Event driven programming.

**Assessment**

Practical exams focused on low level C capabilities, threads in Java, shared memory and message passing. In addition, participation in class and a final homework on Swing and background threads will be part of the final evaluation.

**Lecturer**

Dr. Sergio Gálvez Rojas  
galvez@uma.es  
Room 3.2.28
**CODE**  208
**NAME**  Networks and Distributed Systems
**Credits**  6 ECTS
**Period**  Spring Semester

**Course Specifications**
Lectures and problem solving sessions; individual and laboratory work.

**Objectives and contents**
This is the first course about Communication Networks with focus on communication protocols covering from physical level thru the application level.

Contents include the following topics: Internet layered network structure. Basic protocol functions such as addressing, multiplexing, routing, forwarding, flow control, re-transmission error recovery schemes, and congestion control. Overview of link, network and transport layer protocol standards, following a bottom up approach. Introduction to wireless and mobile networks. This course will also give hands-on experience in network programming using the socket API in C and Java programming languages.

**Assessment**
The acquisition of concepts is evaluated considering: class attendance and participation; midterm and final exams (70% aprox.). The practical part of this course counts 30% of the final grade.

**Lecturer**
Dr. Lidia Fuentes
lf@lcc.uma.es
Room 3.2.2

**CODE**  209 (305)
**NAME**  Intelligent Systems
**Credits**  6 ECTS
**Period**  Spring Semester (Fall Semester)

**Course Specifications**
Lectures, written exercises, individual work, programming laboratory practices.

**Objectives and contents**
This is a first course in Artificial Intelligence (AI). AI is ‘the art of creating machines that perform functions that require intelligence when performed by people’ (Kurzweil, 1990). We offer a broad view of the problems that AI can solve. The aim is to provide a general knowledge of this branch of Computer Science, so that you are able to take more advanced courses on specific subfields. Course contents include: search, games, logic, planning, artificial neural networks and decision problems.

**Assessment**
There are two written exams: a midterm exam and a final exam. The grade of the midterm exam is used to improve the grade of the final exam, in case that the grade of the final exam is lower. Also, there are mandatory programming exercises, to be coded in Java. Therefore, programming skill is a prerequisite. Active class participation and other optional activities give extra marks.

**Lecturers**
Dr. Ezequiel López-Rubio
ezeqlr@lcc.uma.es
Room: 3.2.42

Dr. José Luis Pérez de la Cruz Molina
perez@lcc.uma.es
Room: 3.2.24
Courses Syllabi

CODE  210

NAME  Operating Systems

Credits  6 ECTS

Period  Spring Semester

Course Specifications
Lectures, exercises, individual development work and laboratory.

Objectives and contents
The focus is on the interaction between the user of a computer and the hardware as handled by an operating system (OS). The student gets a feeling how an OS manages all the resources, i.e. processor, memory and IO and how it provides an interface between an application program and the computer hardware by handling

Contents:
- Processes and threads
- Process scheduling
- Memory management
- File systems

Assessment
During the course the student studies a book, makes and hands in exercises, visits a common laboratory and hands in defends individual design assignments. There are two theoretical midterm exams that count for 50% of the final mark. The handed in practical work counts for the other 50% of the final mark. A final theoretical exam will also be scheduled for students who did not pass one of the midterm exams.

Lecturers
Dr. Guillermo Pérez Trabado
gperez@uma.es
Room 2.2.34

Dr. Eligius M.T. Hendrix
eligius@uma.es
https://sites.google.com/site/eligiushendrix/
Full year offer (87 ECTS)

<table>
<thead>
<tr>
<th>Fall Semester</th>
<th>Spring Semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fundamentals of Programming (6)</td>
<td>Object Oriented Programming (6)</td>
</tr>
<tr>
<td>Analysis and Design of Algorithms (6)</td>
<td>Computer Technology (6)</td>
</tr>
<tr>
<td>Databases (6)</td>
<td>Introduction to Software Engineering (6)</td>
</tr>
<tr>
<td>Computer Organization (6)</td>
<td>Systems Program and Concurrency (6)</td>
</tr>
<tr>
<td>Data Structures (6)</td>
<td>Networks and Distributed Systems (6)</td>
</tr>
<tr>
<td>Computational Intelligence (4.5)</td>
<td>Operating Systems (6)</td>
</tr>
<tr>
<td>Biomedical Computational Modeling (4.5)</td>
<td></td>
</tr>
</tbody>
</table>