

Programming 1

Presentation 2013-2014



```

// Calcula la Nota de FP1 del Curso 2009-2010
#include <iostream>
using namespace std;

main()
{
    char convocatoria;
    float ordenador_1, ordenador_2, examen_escrito;
    float nota_final;

    cout << "Dime la convocatoria(F,J,D):";
    cin >> convocatoria;
    cout << "Dime la nota del examen escrito:";
    cin >> examen_escrito;
    nota_final = examen_escrito;

    cout << "TU NOTA FINAL ES " << nota_final << endl;
}
else
    nota_final = 0.1*ordenador_1 + 0.25*ordenador_2 + 0.65*examen_escrito;
}
if (convocatoria == 'J' || convocatoria == 'D') {
    cout << "Dime la nota del examen escrito:";
    cin >> examen_escrito;
    nota_final = examen_escrito;
}
}
cout << "TU NOTA FINAL ES " << nota_final << endl;
}

```



Dpnt. de Ciència de la Computació i Intel·ligència *a*rtificial
Dpto. de Ciencia de la Computación e Inteligencia *a*rtificial

Teachers

- **Rafael Molina Carmona (rmolina@dccia.ua.es)**
 - Theory classes
 - Office: Politécnica II, 1st floor, orange columns corridor
 - Tutorial hours (send email or CV tutorial)
 - Monday 10:00 - 13:00
 - Friday 10:00 - 13:00
- **Cristina Pomares Puig (cpomares@dccia.ua.es)**
 - Practical classes
 - Office: Politécnica II, 1st floor, green columns corridor
 - Tutorial hours
 - Monday 9:30 - 13:30
 - Wednesday 9:30 - 11:30

Objectives / Competences

1. Analyzing problems that can be solved using a computer and designing algorithms to solve them
 - **Building algorithms regardless of the programming language**
2. Implementing algorithms using structured programming techniques
 - **Knowing and using a high level programming language (C language)**
3. Being constant and responsible in work
 - **Individual work and/or group work**
 - **Constant work**

Problem solving and programming

It's the only job I can think of where I get to be both an engineer and an artist. There's an incredible, rigorous, technical element to it, which I like because you have to do very precise thinking. On the other hand, it has a wildly creative side where the boundaries of imagination are the only real limitation.

*Andy Hertzfeld, co-designer of Macintosh,
in *Programmers at Work**

Syllabus

1. Introduction (1 session)
2. Simple Data Types (1 session)
3. Control Statements (2 sessions)
4. Modular Programming (3 sessions)
5. Recursion (2 sessions)
6. Structured Data Types: Arrays (2 sessions)
7. Structured Data Types: Structs (2 sessions)
8. Evaluating the Temporary Cost of an Algorithm (1 session)

Theory classes planning

Date	Activity
12-Sep	Lecture 1
19-Sep	Lecture 2/Control
26-Sep	Lecture 3
3-Oct	Lecture 3
10-Oct	Lecture 4
17-Oct	Lecture 4
24-Oct	Lecture 4

Date	Activity
31-Oct	Lecture 5
7-Nov	Lecture 5/Exercises
14-Nov	Lecture 6
21-Nov	Lecture 6
28-Nov	Lecture 7
5-Dec	Lecture 7
12-Dec	Lecture 8
19-Dec	Revision

Practice

1. Introduction to Linux. My first program in C (1 session)
2. Simple Data Types (1 session)
3. Conditional Control Statements (2 sessions)
4. Iterative Control Statements (1 session)
5. Modular Programming (2 sessions)
6. Recursion (1 session)
7. Structured Data Types: Arrays (2 sessions)
8. Structured Data Types: Structs (2 sessions)

Practical classes planning

Date	Activity
11-Sep	Presentation / Practice 1
18-Sep	Practice 2
25-Sep	Practice 3
2-Oct	Practice 3
9-Oct	
16-Oct	Practice 4
23-Oct	Control/Practice 5

Date	Activity
30-Oct	Practice 5
6-Nov	Practice 6
13-Nov	Partial exam 1
20-Nov	Practice 7
27-Nov	Practice 7
4-Dec	Practice 8
11-Dec	Practice 8
18-Dec	Partial exam 2

Access to the following practical exercise will be enabled once the marked exercise in the previous practical exercise is handed in.

Evaluation

- **January exams**

If ($WE \geq 4$ and $PE2 \geq 4$) then

$$\text{Mark} = 0.15*PE1 + 0,15*CONTR + 0.35*PE2 + 0.35*WE$$

else

$$\text{Mark} = \text{minimum}(\text{Mark}, 4.5)$$

- **July exams**

The written exam and the practical exam 2 can be retaken.

The controls and the practical exam 1 cannot be retaken.

WE = Written Exam

PE = Practical Exam (with the computer)

CONTR = Controls

Materials

- *On Campus Virtual (Moodle)*
 - Objectives, planning, activities and evaluation
 - Slides
 - Instructions for practical work
 - Interesting internet links
- **DVD with the laboratory software**
 - To install Linux Ubuntu in a virtual machine on your personal computer
 - To do the practical work at home

Reference Bibliography

Fundamentos de Programación

Jesús Carretero, Félix García, y otros

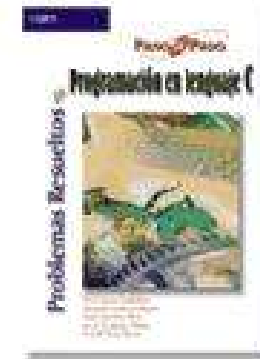
Thomson-Paraninfo (2007) ISBN: 978-84-9732-550-9



Problemas Resueltos de Programación en Lenguaje C

Félix García, Alejandro Calderón, y otros

Thomson (2002) ISBN: 84-9732-102-2



Resolución de Problemas con C++

Walter Savitch

Pearson Addison Wesley (2007) ISBN: 978-970-26-0806-6



Source of Basic Information

Google

YAHOO!



bing™



Advice about classes

- **Theory class attendance**
 - Every day attendance
 - Take notes
 - Ask the teacher your doubts
- **Practical class attendance**
 - Every day attendance (compulsory, maximum 2 absences allowed)
 - Do the practical exercises
 - If you don't know how to do an exercise or have doubts, follow these steps:
 - Review the theory topics
 - Search in the recommended books or on Internet
 - Discuss the problem with your classmates
 - Ask the teacher your doubts

Advice about homework

- **Studying at home**
 - During the same week of the theory class
 - Complete the slides and your notes. Use the recommended books and internet
 - Share and discuss your doubts with your classmates
 - Ask the teacher in his tutorial hours or by *Campus Virtual*
- **Doing the practical work at home**
 - Read the instructions for the practical work before the class
 - Do the exercises that you couldn't finish in the laboratory
 - Write down the doubts so you can share them with your classmates and teacher in the next class
 - If you get stuck with a problem, search the information in books and on Internet
 - Ask the teacher in his tutorial hours or by *Campus Virtual*

From our experience

- If you don't understand or work on the theory, you won't understand or work on the practical part
- If you don't understand or work on the practical part, you won't pass the exams

Therefore

It is almost impossible to succeed if
You just study the last days
or
You don't do the practical work

The key: constant work