



UNIVERSIDAD
DPTO. DE LENGUAJES Y C. DE LA COMPUTACION
E.T.S. DE INGENIERIA INFORMATICA

FUNDAMENTOS DE LA PROGRAMACIÓN

(Primer Cuatrimestre. Créditos 6)

Objetivos: Esta asignatura debe aportar al alumno los elementos de programación básicos, de una manera rigurosa y estructurada, haciendo especial hincapié en una buena metodología de programación para la resolución de problemas. Iniciará al alumno en las técnicas de programación estructurada, en el uso del diseño descendente y la abstracción procedimental.

Metodología: La metodología de trabajo girará en torno a los siguientes bloques de actividades formativas:

- Clases magistrales de introducción y desarrollo de la teoría que compone la asignatura.
- Clases prácticas en pizarra sobre realización de problemas y ejercicios.
- Actividades en laboratorio
- Trabajos en grupo o individuales.
- Tutorías en grupo e individuales.
- Estudio autónomo del alumno de los contenidos teóricos y prácticos de la asignatura.

En las clases de teoría el alumno irá conociendo los diferentes conceptos de la asignatura, que serán llevados a la práctica tanto en las clases de problemas en pizarra como en las clases de prácticas en laboratorio, con la utilización del lenguaje de programación C++ y el entorno de desarrollo Eclipse. Para las clases de laboratorio se formarán grupos reducidos de alumnos con objeto de conseguir una interacción profesor-alumno más estrecha. Por otro lado, el alumno debe afianzar todos los conocimientos y prácticas adquiridos en las clases presenciales mediante el estudio y la resolución de problemas de forma autónoma.

Cada tema estará disponible en el servicio de reprografía del centro y en la página web de la asignatura dentro del Campus Virtual de la UMA (<http://www.informatica.cv.uma.es>) antes de que se imparta.

Evaluación: A lo largo del curso se realizarán controles periódicos con objeto de estimar el progreso y la adquisición de conocimientos por parte del alumno.

La calificación final de un alumno será la suma ponderada de la calificación obtenida en cada uno de los siguientes bloques, siendo necesario alcanzar una calificación mínima en el bloque 2.

- Bloque 1 [30%] Seguimiento del trabajo continuado por parte del alumno mediante su participación activa en clase, la realización de los trabajos establecidos y la realización de controles periódicos.
- Bloque 2 [70%] Realización de un examen final de la asignatura completa.

Según establecido en el POD aprobado en el Consejo de Gobierno del 7 de Mayo de 2010, la asistencia de los alumnos a clase es obligatoria. Sin embargo, se contempla la posibilidad de que aquellos alumnos a los que, por motivos justificados, les sea imposible cumplir este requisito, superen la asignatura. Para ello deberán contactar con el profesor al inicio del curso y cumplir el plan de trabajo establecido (tareas y plazos) y superar las pruebas evaluativas. El citado plan de trabajo se difundirá y se llevará acabo a través del Campus Virtual

Programa:

I. INTRODUCCIÓN A LA PROGRAMACIÓN

- I.1. La Informática y el Papel de la Programación de Computadores.
- I.2. El Computador: una Máquina que Procesa Información.
- I.3. Codificación de la Información.
 - I.3.1. Representación posicional de los números.
 - I.3.2. Códigos de E/S.
- I.4. Estructura Funcional de los Computadores.
 - I.4.1. Funcionamiento interno de los Computadores.
- I.5. Algoritmos y Resolución de Problemas.
 - I.5.1. ¿Qué se quiere hacer? Concepto de Algoritmo.
 - I.5.2. ¿Qué se puede hacer? Calculabilidad y Complejidad.
 - I.5.3. ¿Cómo hay que hacerlo? Corrección.
- I.6. Lenguajes de Programación
 - I.6.1. Reconocimiento de Lenguajes. Gramáticas.
 - I.6.2. Traductores, Compiladores e Interpretes.
- I.7. Visión General de un Sistema Informático
 - I.7.1. Entorno integrado de desarrollo.

II. INTRODUCCIÓN A UN LENGUAJE DE PROGRAMACIÓN

II.1. Introducción a C++.

II.1.1. La evolución de C++.

II.1.2. Ejemplo de programa en C++.

II.1.3. Elementos básicos de C++.

II.1.4. Esquema general de un programa en C++.

II.2. Tipos de datos simples.

II.2.1. Tipos simples predefinidos.

II.2.2. Tipos simples definidos por el programador.

II.2.3. Operadores.

II.2.4. Conversiones de tipos.

II.3. Constantes, variables y asignaciones.

II.4. Entrada y salida básicas.

II.5. Flujo de control.

II.6. Expresiones Lógicas o Booleanas.

II.7. Estructuras de selección.

II.7.1. Estructura *If*.

II.7.2. Estructura *switch*.

II.8. Estructuras de iteración.

II.8.1. Estructura *while*.

II.8.2. Estructura *do while*.

II.8.3. Estructura *for*.

II.8.4. Diseño de bucles. Concepto de Invariante.

II.9. Control de errores y excepciones.

II.10. Recomendaciones y errores frecuentes.

III. ABSTRACCIÓN PROCEDIMENTAL

III.1. Diseño Descendente.

III.2. Procedimientos y Funciones. Parámetros.

III.2.1. Ejemplo.

III.2.2. Declaración de Procedimientos y Funciones. Parámetros Formales.

III.2.3. Llamada a Procedimientos y Funciones. Parámetros Reales.

III.2.4. Paso de parámetros por valor y por referencia.

III.2.5. Interfaz.

III.2.6. Criterios de modularización.

III.2.7. Variables locales y globales. Efectos laterales.

III.2.8. Precondiciones y Postcondiciones. Tratamiento de situaciones excepcionales.

III.3. Recursividad.

III.3.1. Concepto de Recursividad.

III.3.2. Ejemplos.

III.3.3. Recursividad frente a Iteración.

IV. TIPOS DE DATOS ESTRUCTURADOS

IV.1. Registros.

IV.1.1. Tipo de Datos *struct*.

IV.1.2. Registros como Parámetros.

IV.2. Arrays.

IV.2.1. Tipo de Datos Array.

IV.2.2. Arrays Multidimensionales.

IV.2.3. Arrays como Parámetros.

IV.3. Cadenas de caracteres.

IV.3.1. Tipo de Datos *string*.

IV.3.2. Cadenas como Parámetros.

IV.4. Ejemplos.

Bibliografía:

Joyanes Aguilar, Luis. Programación en C++. Algoritmos, estructuras de datos y objetos. McGraw Hill, 2ª Edición 2006.

Savitch, Walter. Resolución de problemas con C++. Pearson Educación, 5ª Edición 2007.