



DESCRIPCIÓN DE LA ASIGNATURA

Grado/Máster en:	Graduado/a en Ingeniería del Software por la Universidad de Málaga
Centro:	Escuela Técnica Superior de Ingeniería Informática
Asignatura:	Fundamentos de la Programación
Código:	104
Tipo:	Formación básica
Materia:	Informática
Módulo:	Formación básica
Experimentalidad:	69 % teórica y 31 % práctica
Idioma en el que se imparte:	Inglés, Español
Curso:	1
Semestre:	1
Nº Créditos	6
Nº Horas de dedicación del estudiante:	150
Nº Horas presenciales:	60
Tamaño del Grupo Grande:	72
Tamaño del Grupo Reducido:	30
Página web de la asignatura:	http://informatica.cv.uma.es

EQUIPO DOCENTE

Departamento: LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

Área: LENGUAJES Y SISTEMAS INFORMÁTICOS

Nombre y Apellidos	Mail	Teléfono Laboral	Despacho	Horario Tutorías
Coordinador/a: BARTOLOME RUBIO MUÑOZ	brubio@uma.es	952132753	3.2.45 - E.T.S.I. INFORMÁTICA	Todo el curso: Viernes 09:00 - 10:30, Martes 09:00 - 10:30, Miércoles 09:00 - 10:30, Lunes 09:00 - 10:30
VICENTE JESUS BENJUMEA GARCIA	vjbenjumea@uma.es	952132754	3.2.3 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Lunes 16:30 - 18:30, Viernes 10:45 - 12:45, Miércoles 12:30 - 14:30 Segundo cuatrimestre: Lunes 15:15 - 17:15, Viernes 15:15 - 17:15, Miércoles 15:15 - 17:15
JUAN ANTONIO FALGUERAS CANO	jfalgueras@uma.es	952132792	3.2.32 - E.T.S. Ing. Telecomunicación	Todo el curso: Lunes 09:00 - 11:00, Miércoles 12:45 - 14:45, Martes 10:00 - 12:00
MANUEL ROLDAN CASTRO	mroldanc@uma.es	952132819	3.2.3 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Lunes 12:00 - 13:00, Lunes 09:30 - 11:30, Miércoles 11:30 - 13:00, Martes 13:00 - 14:30 Segundo cuatrimestre: Martes 10:00 - 11:30, Viernes 13:00 - 14:30, Miércoles 09:00 - 11:00, Martes 13:00 - 14:00

RECOMENDACIONES Y ORIENTACIONES

Se parte de cero, por lo que no es necesario ningún conocimiento previo sobre los contenidos de la asignatura por parte del alumno.

CONTEXTO

La asignatura es la primera de las que conforman el grupo de asignaturas del grado dedicadas a la programación de ordenadores. Constituye un primer acercamiento del alumno al mundo de la programación y debe aportar al mismo 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.

COMPETENCIAS

1 Competencias generales y básicas.

BÁSICAS

- CB02** Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
- CB03** Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
- CB04** Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
- CB05** Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

GENERALES

- CG08** Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.



1 Competencias generales y básicas.

GENERALES

- CG09** Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática

2 Competencias específicas.

Formacion Basica

- CE-CB03** Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.
- CE-CB04** Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
- CE-CB05** Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.

CONTENIDOS DE LA ASIGNATURA

Único

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 Intérpretes.
- 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. Definición y 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.1.3. Ejemplo con registros
- IV.2. Arrays.
 - IV.2.1. Tipo de Datos Array.
 - IV.2.2. Arrays como Parámetros.
 - IV.2.3. Problemas comunes al trabajar con arrays.
 - IV.2.4. Utilidad de los arrays.
 - IV.2.5. Arrays Multidimensionales.
 - IV.2.6. Arrays Abiertos como Parámetros.
- IV.3. Cadenas de caracteres.
 - IV.3.1. Tipo de Datos string.
 - IV.3.2. Strings como Parámetros.
- IV.4. Resolución de Problemas usando Tipos Estructurados.



Actividades presenciales

Actividades expositivas

Lección magistral

Actividades prácticas en instalaciones específicas

Prácticas en laboratorio

Actividades no presenciales

Actividades prácticas

Resolución de problemas

Estudio personal

Estudio personal

ACTIVIDADES DE EVALUACIÓN

Actividades de evaluación presenciales

Actividades de evaluación del estudiante

Examen parcial

Examen final

RESULTADOS DE APRENDIZAJE / CRITERIOS DE EVALUACIÓN

Lo que se espera que un estudiante conozca, comprenda y sea capaz de hacer tras cursar esta asignatura queda reflejado en los siguientes Resultados de Aprendizaje:

- RA1 - Plantear y diseñar soluciones algorítmicas a problemas concretos mediante el uso de la programación estructurada y la abstracción procedimental.
- RA2 - Valorar la importancia de la abstracción, especialmente con respecto a la programación de sistemas de cierta envergadura.
- RA3 - Escribir programas imperativos aplicando tipos de datos simples y estructurados, estructuras de selección e iteración y la recursividad.
- RA4 - Utilizar entornos y herramientas de desarrollo con los que implementar los algoritmos diseñados con el lenguaje de programación C++.
- RA5 - Identificar, localizar y corregir los errores que puedan aparecer en las soluciones obtenidas para los problemas planteados.

Estos RA desarrollan claramente las Competencias Específicas fijadas. En concreto, se desarrolla:

- La capacidad para comprender y dominar los conceptos básicos de algorítmica y su aplicación para la resolución de problemas (CE-CB03)
- Los conocimientos básicos sobre programación de los ordenadores (CE-CB04)
- El conocimiento de la estructura, organización y funcionamiento de los sistemas informáticos, los fundamentos de su programación y su aplicación para la resolución de problemas (CE-CB05)

Por otro lado, las Competencias Generales establecidas para la asignatura quedan también cubiertas con estos RA. Concretamente:

- La competencia CG08 está cubierta fundamentalmente por RA2 y RA4.
- La competencia CG09 queda cubierta por RA1, RA3 y RA5.

Tanto para la realización de las prácticas de laboratorio como la resolución de los problemas propuestos a lo largo de la asignatura los estudiantes aplicarán sus capacidades y conocimientos adquiridos conforme a lo reflejado en las Competencias Básicas establecidas (CB02 - CB05).

Por su parte, las actividades de evaluación establecidas (exámenes parciales y examen final, con una componente teórica y otra práctica), junto con el procedimiento de evaluación elegido, claramente sirven para evaluar si los estudiantes consiguen alcanzar los RA.

PROCEDIMIENTO DE EVALUACIÓN

A lo largo del curso los alumnos entregarán las prácticas que se propongan. Además, se realizarán controles periódicos (cuyas fechas serán previamente conocidas por los alumnos) con objeto de estimar el progreso y la adquisición de conocimientos por parte del alumno.

Los alumnos que hayan asistido y entregado al menos el 60% de las prácticas se considerarán que siguen un proceso de evaluación continua.

Para los alumnos que siguen el proceso de evaluación continua, la calificación en las convocatorias ordinarias de Febrero y Septiembre se realiza de la siguiente forma:

- Se califican los controles realizados durante el curso sobre un máximo de 3 puntos.
- Se califica un examen final con un máximo de 10 puntos menos lo obtenido en el punto anterior.

Para los alumnos que no siguen el proceso de evaluación continua, la calificación en las convocatorias ordinarias de Febrero y Septiembre se realiza de la siguiente forma:

- Se califica un examen final sobre un máximo de 7 puntos.



- Se califica un cuestionario teórico sobre un máximo de 3 puntos.

En ambos casos, se suman las dos calificaciones y la calificación mínima para aprobar es de 5 puntos.

Para las convocatorias extraordinarias sólo existirá un único examen final que puntúa el 100% de la calificación.

El profesor se reserva el derecho de verificar la calificación obtenida por un alumno, tanto en los controles como en los exámenes finales, realizando la correspondiente revisión en presencia del mismo.

El plagio de cualquier actividad de la asignatura supondrá una calificación de 0 puntos para los implicados en la convocatoria correspondiente.

Los alumnos a tiempo parcial y los alumnos deportistas de alto nivel universitarios se considerarán en principio alumnos que no siguen el proceso de evaluación continua. No obstante, en función de su situación particular, pueden optar por el proceso de evaluación continua.

BIBLIOGRAFÍA Y OTROS RECURSOS

Básica

- Gregorio Rodríguez, Carlos y otros. Ejercicios de Programación Creativos y Recreativos en C++. Prentice Hall, 2002. ISBN: 84-205-3211-8
Joyanes Aguilar, Luis. Programación en C++. Algoritmos, estructuras de datos y objetos. McGraw Hill, 2ª Edición 2006. ISBN-10: 844814645X
Savitch, Walter. Resolución de problemas con C++. Pearson Educación, 5ª Edición 2007. ISBN-10: 970-26-0806-6

DISTRIBUCIÓN DEL TRABAJO DEL ESTUDIANTE

ACTIVIDAD FORMATIVA PRESENCIAL

Descripción	Horas	Grupo grande	Grupos reducidos
Lección magistral	41,4	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Prácticas en laboratorio	18,6	<input type="checkbox"/>	<input checked="" type="checkbox"/>

TOTAL HORAS ACTIVIDAD FORMATIVA PRESENCIAL 60

ACTIVIDAD FORMATIVA NO PRESENCIAL

Descripción	Horas
Resolución de problemas	50
Estudio personal	25

TOTAL HORAS ACTIVIDAD FORMATIVA NO PRESENCIAL 75

TOTAL HORAS ACTIVIDAD EVALUACIÓN 15

TOTAL HORAS DE TRABAJO DEL ESTUDIANTE 150

