



Grado en Ingeniería Informática

DATOS DE LA ASIGNATURA

Nombre:

Fundamentos de Programación

Denominación en inglés:

Fundamentals Programming

Código:

606010104

Carácter:

Básico

Horas:**Totales****Presenciales****No presenciales****Trabajo estimado:**

150

60

90

Créditos:**Grupos reducidos****Grupos grandes****Aula estándar****Laboratorio****Prácticas de campo****Aula de informática**

3

0

0

0

3

Departamentos:**Áreas de Conocimiento:**

Tecnologías de la Información

Lenguaje y Sistemas Informáticos

Curso:**Cuatrimestre:**

1º - Primero

Primer cuatrimestre

DATOS DE LOS PROFESORES

Nombre:**E-Mail:****Teléfono:****Despacho:**

*Martín Ramos, José
Manuel

jmmartin@dti.uhu.es

959217637

TUPB-43

Polo Almohano, M^a del Pilar

polo@dti.uhu.es

87386

nº 53 Edificio Torreumbria

Roche Beltrán, Francisco

roche@dti.uhu.es

959217654

21 de Torreumbria (ETSI)

*Profesor coordinador de la asignatura

Consultar los horarios de la asignatura

1. Descripción de contenidos

1.1. Breve descripción (en castellano):

Introducción a los Lenguajes de Programación

- Concepto e historia de los lenguajes de programación
- Paradigmas de Programación
- Compiladores e Intérpretes.

Introducción a la Programación Orientada a Objetos

Algoritmos y Tipos de Datos.

- Algoritmos. Tipos de Datos, Operadores y Expresiones.
- Estructuras de Control.
- Tipos Básicos de Datos Estructurados.
- Métodos e interfaces.

Técnicas de Diseño de Programas.

- Diseño Descendente.
- Diseño Modular.

1.2. Breve descripción (en inglés):

Introduction to Programming Languages

- Definition and history of programming languages
- Programming Paradigms
- Compilers and Interpreters.

Introduction to Object Oriented Programming

Algorithms and Data Types.

- Algorithms. Data Types, Operators and Expressions.
- Control Structures.
- Basic Types of Structured Data.
- Methods and interfaces.

Program Design Techniques.

- Design Descending.
- Modular Design.

2. Situación de la asignatura

2.1. Contexto dentro de la titulación:

La asignatura de Fundamentos de la Programación es la primera asignatura en la que el alumno aprecia el papel central que representa la abstracción en la tarea de programar, a la vez que conoce un lenguaje de programación de alto nivel para codificar los programas, en el paradigma de la programación orientada a objetos. Es por tanto, imprescindible tanto en asignaturas específicas de programación como en aquellas en las que se hace uso de un lenguaje de programación:

- Estructuras de Datos I.
- Estructuras de Datos II.
- Metodología de la Programación.
- Ingeniería del Software.
- Sistemas Operativos.
- Programación Concurrente.

2.2. Recomendaciones:

El alumno debe estudiar la asignatura consultando la bibliografía sugerida por los profesores y asistir con regularidad a las tutorías que estos ofertan. Debido al marcado carácter práctico de la asignatura, se recomienda que el alumnado realice la mayor parte de los ejercicios propuestos de las relaciones de problemas.

Los alumnos que cursen esta asignatura deberían :

- Saber leer, comprender y expresarse correctamente tanto de forma oral como escrita.
- Tener conocimientos matemáticos a nivel de Bachillerato.
- Conocer el idioma inglés tanto a nivel escrito como oral (nivel medio).

3. Objetivos (Expresados como resultados del aprendizaje):

- Desarrollar una aproximación disciplinada a la especificación, implementación, verificación y documentación de programas.
- Aprender el papel central que representa la abstracción en la tarea de programar
- Desarrollar en el alumno la capacidad de resolución de problemas mediante las técnicas de diseño de algoritmos y aplicarlo a la codificación de programas
- Conocer y utilizar adecuadamente estructuras de datos básicas, algoritmos y esquemas de uso general
- Aprender un lenguaje de programación de alto nivel estructurado, general y extendido que use el paradigma de la programación orientado a objetos.
- Proporcionar los fundamentos teóricos y prácticos básicos para cursar posteriores estudios en programación.

4. Competencias a adquirir por los estudiantes

4.1. Competencias específicas:

- **CB04:** Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería
- **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.

4.2. Competencias básicas, generales o transversales:

- **CB1:** Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio
- **CB5:** Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía

5. Actividades Formativas y Metodologías Docentes

5.1. Actividades formativas:

- Sesiones de Teoría sobre los contenidos del Programa.
- Sesiones Prácticas en Laboratorios Especializados o en Aulas de Informática.

5.2. Metodologías docentes:

- Clase Magistral Participativa.
- Desarrollo de Prácticas en Laboratorios Especializados o Aulas de Informática en grupos reducidos.
- Resolución de Problemas y Ejercicios Prácticos.
- Evaluaciones y Exámenes.

5.3. Desarrollo y justificación:

Sesiones académicas de teoría

- Consisten en clases magistrales donde se impartirá la base teórica de la asignatura y se expondrán ejemplos aclaratorios de la misma al grupo. La sesión teórica tendrá una duración de 1h, dos veces por semana, sumando un total de 30h y se irán intercalando con las sesiones de problemas a lo largo del curso, de manera que una vez finalizado un tema teórico con sus correspondientes sesiones académicas de teoría, se impartirán sesiones de problemas.
- La metodología usada para impartir la teoría y los ejemplos aclaratorios será la exposición mediante cañón, retroproyector y/o uso de pizarra.

Sesiones prácticas en laboratorio / Resolución y entrega de prácticas

- Consisten en el diseño e implementación de programas escritos en C++ utilizando orientación a objetos desde el primer día. Los alumnos dispondrán con antelación del problema a resolver y la metodología de trabajo. El trabajo se realizará de forma individual. Las sesiones prácticas tendrán una duración de 2h cada una, sumando un total de 30h. La asistencia a las sesiones de laboratorio es obligatoria.

Realización de pruebas parciales evaluables

- A lo largo del curso se realizarán 2 pruebas parciales evaluables sobre las prácticas en laboratorio. La primera prueba parcial (práctica) se realizará sobre los Temas 1, 2, 3 y 4 y consistirá en realizar una o varias modificaciones a los programas realizados o bien desarrollar un o varios programas nuevos de la complejidad de los propuestos en clase. La segunda prueba parcial (práctica) se realizará sobre el Tema 5, desarrollando una práctica final sobre la que se realizará una prueba práctica de modificación.

6. Temario desarrollado:

TEORIA:

Unidad didáctica nº 1.- Introducción a los Lenguajes de Programación.

TEMA 1. LENGUAJES DE PROGRAMACIÓN

- 1.1. Concepto e historia de los lenguajes de programación.
- 1.2. Paradigmas de Programación.
- 1.3. Clasificación. Lenguaje de bajo, medio y alto nivel.
- 1.4. Traductores: Compiladores e Intérpretes. Proceso de generación de código ejecutable. Compilación y Enlazado.
- 1.5. Programación Orientada a Objetos.

Unidad didáctica nº 2.- Algoritmos y Tipos de Datos.

TEMA 2. ALGORITMOS. Tipos de Datos, Operadores y Expresiones

- 2.1. Concepto de algoritmo. Estructura general.
- 2.2. Palabras clave, Identificadores, constantes y comentarios. Variables y objetos.
- 2.3. Tipos de datos y Clases. Clasificación.
- 2.4. Operadores de asignación, aritméticos, relacionales y lógicos.
- 2.5. Expresiones y orden de precedencia.
 - 2.5.1. Reglas de precedencia y asociatividad.
 - 2.5.2. Conversiones de tipos en las expresiones.
 - 2.5.3. Coherencia de tipos. Moldeado de tipos.
- 2.6. Operaciones básicas de entrada-salida.

TEMA 3. ESTRUCTURAS DE CONTROL

- 3.1. Sentencia Secuenciales.
- 3.2. Sentencias Condicionales.
- 3.3. Sentencias Iterativas.
- 3.4. Macros de sentencias.

TEMA 4. TIPOS DE DATOS ESTRUCTURADOS

- 4.1. Registros y Estructuras de registros. Registros jerárquicos.
- 4.2. Vectores y Matrices. Vectores de Registros
- 4.3. Clases y Objetos.
- 4.4. Declaraciones privadas y públicas. Ámbitos de visibilidad.

Unidad didáctica nº 3.- Técnicas de diseño de programas.

TEMA 5. DISEÑO DESCENDENTE

- 5.1. Programación Estructurada.
- 5.2. Declaraciones globales y declaraciones locales. Ámbitos de visibilidad.
- 5.3. Funciones y Métodos: Interfaces.
- 5.4. Parametrización. Parámetros por valor y por referencia.
- 5.5. Constructores y destructores.
- 5.6. Paso de estructuras de datos complejos a funciones y métodos.
- 5.7. Sobrecarga de métodos y operadores.

TEMA 6. DISEÑO MODULAR

- 6.1. Programación Modular.
- 6.2. Concepto de módulo.
- 6.3. Creación y utilización de módulos
- 6.4. Importación y exportación.
- 6.5. Definición y uso de Módulos.

LABORATORIO:

Unidad didáctica nº 2.- Algoritmos y Tipos de Datos.

Prácticas 1, 2 y 3. Introducción a C++, uso de clases. Uso de cadenas y tablas.

Unidad didáctica nº 3.- Técnicas de diseño de programas.

Práctica 4. Diseño descendente.

7. Bibliografía

7.1. Bibliografía básica:

- METODOLOGÍA DE LA PROGRAMACIÓN I: INTRODUCCIÓN AL DISEÑO ORIENTADO A OBJETOS. A. Márquez, Lourdes Ortiz, M^a Pilar Polo, Fco. Roche y Ana M^a Roldán. Servicio de Publicaciones de la Universidad de Huelva.
- COMO PROGRAMAR EN C/C++. H.M. Deitel. Edt. PEARSON Prentice Hall.
- C++ ESTÁNDAR. E. Hernández Orallo. Edt. Paraninfo, Thomson Learning.
- PROGRAMACIÓN EN C++ PARA INGENIEROS. F. Xhafa, P. Vázquez, J. Marco, X. Molinero y A. Martín. Edt. Thomson.
- EL LENGUAJE DE PROGRAMACIÓN C++, B. Stroustrup. Ed. PEARSON Addison Wesley.
- PROGRAMACIÓN Y DISEÑO EN C++, J.P. Cohoon, J.W. Davidson. Edt. Mcgraw-Hill

7.2. Bibliografía complementaria:

- RESOLUCIÓN DE PROBLEMAS CON C++. W. Savitch. PEARSON Addison Wesley.

8. Sistemas y criterios de evaluación.

8.1. Sistemas de evaluación:

- Examen de teoría/problemas
- Examen de prácticas

8.2. Criterios de evaluación y calificación:

Los criterios de evaluación son:

- **Examen Teórico-Práctico (Teoría).** Consistente en una prueba donde el alumno deberá resolver distintos problemas y/o cuestiones teóricas referentes a los temas teóricos desarrollados durante el cuatrimestre. Competencias B4 y B5. La puntuación máxima de este criterio es el 50% de la nota final.
- **Examen Práctico en aula de informática (Prácticas).** Una vez desarrolladas las primeras prácticas se realizará una evaluación sobre el contenido de las mismas. Así mismo en la última sesión de clase se realizará otra prueba consistente en la modificación de la práctica final que haya realizado el alumno. Competencias B4 y B5. La puntuación máxima de este criterio es del 50% de la nota final. La práctica final tendrá un peso del 70% del criterio Práctico (3.5 puntos) y el resto de prácticas el (30%) del criterio Práctico (1.5 puntos).

Los criterios de calificación para las convocatorias Febrero y Septiembre son:

- La puntuación de cada criterio (Teoría y Prácticas) es de 5 puntos (50% de la nota final) como máximo.
- Para que cada criterio sea contabilizado en la Nota Final de la asignatura (Nota de Acta) se ha de obtener una puntuación mínima de 2 puntos (40% de 5 puntos).
- La fórmula de obtener la nota Final de la asignatura, tanto en la convocatoria de Febrero como en la de Septiembre, se resume en la siguiente regla:
SI Examen Teórico \geq 2 puntos (sobre 5) y Prácticas \geq 2 puntos (sobre 5)
Nota Acta = Examen Teórico + Prácticas
en caso contrario
Nota Acta = Nota más baja de las obtenidas entre el Examen Teórico y las Prácticas.

Criterio para guardar las notas para convocatoria de Septiembre:

- La nota del examen Teórico así como la nota de las Prácticas podrán ser guardadas para la convocatoria de Septiembre siempre y cuando se obtenga un mínimo de 2 puntos (40%) sobre 5.
- El alumno podrá presentarse en la convocatoria de Septiembre, tanto al examen práctico como al examen teórico, independientemente de que le sean guardadas o no sus notas.
- Si tiene guardada una nota de un criterio (Teoría o Prácticas) y se presenta a subir nota, para el cálculo de la nota final se utilizará la última puntuación obtenida y no la guardada.
- Como se ha mencionado anteriormente, el cálculo de la nota Final se realizará mediante la misma regla utilizada en la convocatoria de Febrero (incluyendo sus mínimos)

9. Organización docente semanal orientativa:

	Semanas	Grupos Grandes	Grupos Reducidos Aula Estándar	Grupos Reducidos Aula de Informática	Grupos Reducidos Laboratorio	Grupos Reducidos prácticas de campo	Pruebas y/o actividades evaluables	Contenido desarrollado
#1	2	0	2	0	0		Presentación y Tema 1	
#2	2	0	2	0	0		Tema 2	
#3	2	0	2	0	0		Tema 2 y 3	
#4	2	0	2	0	0		Tema 3	
#5	2	0	2	0	0		Tema 3 y 4	
#6	2	0	2	0	0		Tema 4	
#7	2	0	2	0	0		Tema 4	
#8	2	0	2	0	0		Tema 5	
#9	2	0	2	0	0		Tema 5	
#10	2	0	2	0	0	Prueba Práctica 1	Tema 5	
#11	2	0	2	0	0		Tema 5	
#12	2	0	2	0	0		Tema 5	
#13	2	0	2	0	0		Tema 5	
#14	2	0	2	0	0		Tema 5	
#15	2	0	2	0	0	Prueba Práctica Final	Tema 6	
	30	0	30	0	0			