

DATOS DE LA ASIGNATURA

Denominación: METODOLOGÍA DE LA PROGRAMACIÓN

Código: 101381

Plan de estudios: GRADO DE INGENIERÍA INFORMÁTICA

Curso: 1

Denominación del módulo al que pertenece: FORMACIÓN BÁSICA

Materia: INFORMÁTICA

Carácter: BASICA

Duración: SEGUNDO CUATRIMESTRE

Créditos ECTS: 6

Horas de trabajo presencial: 60

Porcentaje de presencialidad: 40%

Horas de trabajo no presencial: 90

Plataforma virtual: <http://www.uco.es/moodle>

DATOS DEL PROFESORADO

Nombre: LUQUE RODRÍGUEZ, MARÍA

Centro: ESCUELA POLITÉCNICA SUPERIOR

Departamento: INFORMÁTICA Y ANÁLISIS NUMÉRICO

Área: CIENCIA DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIA

Ubicación del despacho: Edificio C2, 3ª Planta

e-Mail: in1lurom@uco.es

Teléfono: 957212257

Nombre: FERNÁNDEZ CABALLERO, JUAN CARLOS

Centro: ESCUELA POLITÉCNICA SUPERIOR

Departamento: INFORMÁTICA Y ANÁLISIS NUMÉRICO

Área: CIENCIA DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIA

Ubicación del despacho: Anexo Edificio C3, Planta Baja

e-Mail: jfcaballero@uco.es

Teléfono: 957218153

Nombre: FERNANDEZ GARCIA, NICOLAS LUIS

Centro: ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA

Departamento: INFORMÁTICA Y ANÁLISIS NUMÉRICO

Área: CIENCIA DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIA

Ubicación del despacho: EDIFICIO ALBERT EINSTEIN, TERCERA PLANTA, PASILLO SUR (CAMPUS DE RABANALES)

e-Mail: ma1fegan@uco.es

Teléfono: 957218347

URL web: www.uco.es/users/ma1fegan

Nombre: GIBAJA GALINDO, EVA LUCRECIA

Centro: ESCUELA POLITÉCNICA SUPERIOR

Departamento: INFORMÁTICA Y ANÁLISIS NUMÉRICO

Área: CIENCIA DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIA

Ubicación del despacho: Edificio C2, 3ª Planta

e-Mail: in1gigae@uco.es

Teléfono: 957212031

Nombre: ZAFRA GÓMEZ, AMELIA

Centro: ESCUELA POLITÉCNICA SUPERIOR

Departamento: INFORMÁTICA Y ANÁLISIS NUMÉRICO

Área: CIENCIA DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIA

Ubicación del despacho: Edificio C2, 3ª Planta

e-Mail: in1zagoa@uco.es

Teléfono: 957212031

DATOS ESPECÍFICOS DE LA ASIGNATURA

REQUISITOS Y RECOMENDACIONES

Requisitos previos establecidos en el plan de estudios

Ninguno.

Recomendaciones

Se recomienda al alumno seguir la asignatura desde el primer día, repasando en casa los conceptos vistos en clase y programando los ejemplos y ejercicios propuestos.

Cualquier ambigüedad en la guía será resuelta por los profesores responsables

COMPETENCIAS

- | | |
|------|--|
| CB4 | Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado. |
| CU2 | Conocer y perfeccionar el nivel de usuario en el ámbito de las TIC |
| CEB4 | Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería |
| CEB5 | 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 |

OBJETIVOS

Proporcionar una importante base teórica y práctica en el aprendizaje de la programación de ordenadores. Esta base permitirá cursar posteriores estudios de programación avanzada.

Concretamente, se pretende que el alumno:

- + Aprecie el papel central que juega la abstracción en la tarea de programar.

- + Maneje correctamente el concepto de recursividad
- + Utilice una metodología basada en el uso de la programación estructurada.
- + Conozca las estructuras de datos, algoritmos y esquemas de uso general.
- + Documente adecuadamente los programas.
- + Realice programas correctos, claros, y eficientes.
- + Verifique el funcionamiento de los programas.
- + Conozca el lenguaje de programación C (lenguaje de alto nivel estructurado, general y extendido), de manera que sea capaz de implementar en dicho lenguaje los algoritmos diseñados previamente y, de este modo, completar adecuadamente el proceso de desarrollo de programas.
- + Sea capaz de entender y seguir el código de los programas realizados por otras personas.
- + Se habitúe a desarrollar programas siguiendo unas normas de estilo tendentes a conseguir programas de calidad. Dentro de estas normas de estilo se engloban aquellas que facilitan la comprensión del código, como pueden ser el uso de comentarios, la indentación de código, ...

CONTENIDOS

1. Contenidos teóricos

- * Ficheros
 - Texto
 - Binario
- * Punteros
- * Estructura de un programa en tiempo de ejecución
 - Memoria dinámica
- * Estructuras lineales dinámicas de datos:
 - Listas
 - Pilas
 - Colas
- * Recursividad
- * Algoritmos básicos de búsqueda y ordenación y su complejidad algorítmica
- * Aspectos metodológicos de la programación
 - Documentación
 - Pruebas
 - Herramientas (makefiles, bibliotecas, depuradores, otras herramientas)

2. Contenidos prácticos

- * Familiarizar al alumno con algún entorno de depuración de programas
- * Aplicar los conceptos de ficheros a programas desarrollados en C
- * Aplicar el concepto de recursividad a la práctica.
- * Gestionar de manera correcta la memoria dinámica.
- * Utilizar el depurador y las bibliotecas.
- * Familiarización con el uso de listas simplemente enlazadas, los algoritmos de ordenación y la herramienta makefile

METODOLOGÍA

Aclaraciones generales sobre la metodología y adaptaciones metodológicas para los alumnos a tiempo parcial

Lecciones magistrales

En estas sesiones se presentarán los conceptos de cada tema a la par que se desarrollarán ejemplos que ayuden al alumno a comprender y clarificar los conceptos. Los ejemplos consistirán en la resolución de algoritmos mediante la aplicación de esquemas de creciente dificultad a lo largo del curso.

Se impartirán en un aula de teoría con la pizarra y el cañón como medios didácticos fundamentales y con el apoyo del ordenador, con el objeto de poder compilar y ejecutar los ejemplos.

La labor del alumno en estas clases consistirá en:

- *Trasladar a sus apuntes las principales ideas que el profesor transmita y preguntar las dudas que le puedan surgir.
- *Participar en la resolución de los ejercicios y problemas propuestos por el profesor.

Una vez en su casa, el alumno debería repasar la clase, comprendiendo los conceptos teóricos y repasando los ejercicios, por si hubiera alguna duda o alternativa en su solución. Estas dudas podrán ser planteadas al profesor en la siguiente clase o más tranquilamente en su horario de tutorías.

Laboratorio

Las clases de prácticas se realizan en un aula de ordenadores con el software necesario para implementar los programas. El objetivo de estas clases consiste en la implementación en el ordenador de un conjunto de ejercicios

de programación cuyo enunciado estará previamente a disposición del alumno en la página web de la asignatura.

Tutorías

Durante el desarrollo de la asignatura, cada alumno dispondrá, además del foro que se colocará en Moodle para la exposición y resolución de dudas de forma colectiva, de las tutorías individualizadas que desee dentro del horario establecido por el profesor. Igualmente, se han programado 2 horas de tutorías colectivas que se celebrarán a lo largo del curso. En estas sesiones se resolverán dudas de teoría de cara al examen final

Resolución de problemas

Se usarán, como apoyo a las clases teóricas y prácticas, y se realizarán de forma individual por cada alumno en el aula. Estas sesiones se centrarán en el análisis de problemas y su planteamiento como algoritmos o programas.

Alumnos a tiempo parcial

Los alumnos a tiempo parcial seguirán la misma metodología que el resto de alumnos

Actividades presenciales

Actividad	Grupo completo	Grupo mediano	Total
<i>Actividades de evaluación</i>	-	2	2
<i>Laboratorio</i>	-	22	22
<i>Lección magistral</i>	26	-	26
<i>Resolución de problemas</i>	8	-	8
<i>Tutorías</i>	2	-	2
Total horas:	36	24	60

Actividades no presenciales

Actividad	Total
<i>Consultas bibliográficas</i>	10
<i>Cuestionarios</i>	2
<i>Ejercicios</i>	47
<i>Estudio</i>	31
Total horas:	90

MATERIAL DE TRABAJO PARA EL ALUMNADO

Cuaderno de Prácticas - <http://www.uco.es/moodle>
Ejercicios y problemas - <http://www.uco.es/moodle>
Enlaces web interesantes - <http://www.uco.es/moodle>
Manual de la asignatura - <http://www.uco.es/moodle>

EVALUACIÓN

Competencias	Instrumentos			
	Programas de intervención	Pruebas objetivas	Resolución de problemas	Pruebas de micros
CB4	x			x
CEB4	x	x	x	x
CEB5		x		
CU2			x	x
Total (100%)	10%	25%	50%	15%
Nota min.(*)	0	5	5	0

(*) Nota mínima necesaria para el cálculo de la media

Periodo de validez de las calificaciones parciales: *Sólo se guardan hasta septiembre las notas de cuestionarios, trabajos y programas de intervención.*

Aclaraciones generales sobre la evaluación y adaptaciones metodológicas para los alumnos a tiempo parcial:

Pruebas objetivas (25%)

*Cuestionarios (5%)

A lo largo del curso, el alumno realizará una serie de cuestionarios sobre los temas impartidos que sólo se valorarán si se han hecho durante el curso y se han entregado en los plazos establecidos. Estos cuestionarios los podrá realizar el alumnos desde casa a través de la plataforma moodle, en el rango de fechas establecido para ello.

*Examen teórico (20%)

Prueba teórica, evaluada de 0 a 10, en el que el alumno debe demostrar sus conocimientos contestando a una serie de preguntas sobre los contenidos teórico/prácticos vistos durante el curso. Si el alumno ha superado todos los cuestionarios propuestos durante el cuatrimestre y su participación es apta, estará exento, si lo desea, de realizar esta prueba obteniendo como calificación la nota media de los cuestionarios. Se debe superar con al menos un 5.

Pruebas de micros (15%)

Durante el cuatrimestre se realizarán dos pruebas de micros en clase de prácticas. Cada una de estas pruebas consistirá en resolver un ejercicio de programación, delante del ordenador, referente a los contenidos vistos en clase hasta el momento.

Resolución de problemas (50%)

Examen sobre el ordenador en el que el alumno deberá realizar diferentes programas en un tiempo determinado, a cuyo término deberán funcionar correctamente. El alumno deberá demostrar que se desenvuelve bien en el desarrollo de programas y de las herramienta: editores, compiladores, depuradores, makefile, etc. Incluirá un problema de nivel medio eliminatorio, de forma que sólo se corregirá el examen si el alumno supera estos conocimientos básicos. Si el alumno ha entregado los análisis de los problemas propuestos en clase de prácticas y ha superado los trabajos de programación, dicho problema no tendrá el carácter de eliminatorio. Se debe superar con al menos un 5 sobre 10.

Programas de Intervención (10%)

Igualmente, se considerará en la evaluación final el grado de participación en clase y en el foro de la asignatura, así como la resolución de ejercicios en pizarra.

Habrá que sacar un 5 entre el examen teórico y la resolución de problemas para sumar el resto de notas

Alumnos a tiempo parcial

Los alumnos a tiempo parcial seguirán el mismo sistema de evaluación que el resto de alumnos.

Convocatoria de septiembre

En la convocatoria de septiembre el alumno deberá recuperar los exámenes teórico y práctico. Una vez realizados los exámenes, se calcularán dos notas, siendo la nota final la más favorable para el alumno.

1) Siguiendo el método de junio

2) 30% examen teórico y 70% examen práctico que podrá contener ejercicio eliminatorio

Convocatoria diciembre

30% examen teórico y 70% examen práctico

Valor de la asistencia en la calificación final:

Criterios de calificación para la obtención de MATRICULA DE HONOR:

BIBLIOGRAFÍA

1. Bibliografía básica:

- * L. Joyanes, I. Zahonero. Programación en C. Metodología, algoritmos y estructuras de datos. McGraw-Hill, 2005.
- * L. Joyanes, A. Castillo, L. Sánchez, I. Zahonero. Programación en C: libro de problemas. McGraw-Hill, 2003.
- * Kernigham, N. B., Ritchie, M. D. El lenguaje de programación C. Prentice-Hall. 1989.

2. Bibliografía complementaria:

Metodología y Algorítmica

- * J.J. García, J.L. Fernández, F.J. Montoya. Una introducción a la programación. Un enfoque algorítmico. Thomson, 2005.
- * F.J. Cortijo, J.C. Cubero, O. Pons. Metodología de la Programación. Programas y estructuras de datos en Pascal. Proyecto Sur, 1993.
- * J.L. Castro et al. Metodología de la Programación. Problemas resueltos. Proyecto Sur, 1993.

Eficiencia y Complejidad

- * J. Fdez-Valdivia, A. Garrido, M. Garcia-Silvente, Estructuras de Datos. Ediciones Gala, 1998.
- * G. Brassard, P. Bratley. Algorítmica, concepción y análisis. Masson, 1996.
- * A. Aho, J.Hopcroft, J. Ullman. Estructuras de datos y algoritmos. Addison-Wesley, 1990

*Galve J.; González J. C.; Sánchez, A. y Velázquez J. A. (1993). Algorítmica. Diseño y análisis de algoritmos funcionales e imperativos. Madrid: editorial Ra-Ma.

Prueba y Verificación-métodos de búsqueda y ordenación

- * L. Joyanes, I. Zahonero. Programación en C. Metodología, algoritmos y estructuras de datos. McGraw-Hill, 2005.

CRITERIOS DE COORDINACIÓN

- Fecha de entrega de trabajos
- Selección de competencias comunes
- Trabajos válidos para varias asignaturas