

ASIGNATURA: 12537 - ESTRUCTURAS DE DATOS II

CENTRO: Escuela de Ingeniería Informática

TITULACIÓN: Ingeniero en Informática

DEPARTAMENTO: INFORMÁTICA Y SISTEMAS

ÁREA: Lenguajes Y Sistemas Informáticos

PLAN: 10 - Año 199**ESPECIALIDAD:**

CURSO: Segundo curso **IMPARTIDA:** Segundo semestre **TIPO:** Troncal

CRÉDITOS: 6

TEÓRICOS: 3

PRÁCTICOS: 3

Descriptores B.O.E.

Estructuras de Datos y Algoritmos de Manipulación. Estructura de Información: Ficheros. Bases de Datos

Temario

MÓDULO 1: PROGRAMACIÓN ORIENTADA A OBJETOS (15 horas, 10T+5P)

Tema 1: Técnicas y metodología de programación orientada a objetos. C++.

Bibliografía básica: [RO99]

Bibliografía complementaria: [ST98], [ME99], [CO00]

MÓDULO 2: ÁRBOLES Y ALMACENAMIENTO (3 horas, 2T+1P)

Tema 2: Ficheros. Reutilización del espacio libre

Tema 3: Árboles binarios. Árbol Trie. Listas invertidas

Bibliografía básica: [HE05]

Bibliografía complementaria: [HE97], [FO92], [FO98]

MÓDULO 3: ÁRBOLES BINARIOS AUTOAJUSTABLES (7 horas, 4T+3P)

Tema 4: Árboles biselados

Tema 5: Árboles AVL

Tema 6: Análisis Amortizado

Tema 7: Árboles Rojo-Negro

Bibliografía básica: [HE05]

Bibliografía complementaria: [HE97], [CO89]

MÓDULO 4: ÁRBOLES MULTIRRAMA (14 horas, 10T+4P)

Tema 8: Árboles multirrama

Tema 9: Árboles B

Tema 10: Árboles B* y B+

Bibliografía básica: [HE05]

Bibliografía complementaria: [FO92]

MÓDULO 5: TÉCNICAS DE DISPERSIÓN (6 horas, 4T+2P)

Tema 11: Dispersión con tabla de tamaño fijo. Resolución de colisiones por: direccionamiento

abierto y mediante encadenamiento
Tema 12: Tablas de dispersión extensible.
Bibliografía básica: [HE05]
Bibliografía complementaria: [HE97], [CO89]

Requisitos Previos

Que el alumno haya cursado con éxito:

- Metodología de la Programación
- Estructuras de Datos I
- Tecnología de la Programación

También es importante tener conocimientos de inglés que permitan la lectura de documentos técnicos.

Objetivos

La preparación y competencia en el manejo de las técnicas y metodologías de la programación orientada a objetos.

La adquisición de destreza en el uso del lenguaje de programación C++.

La utilización y el manejo de los ficheros como medios de almacenamiento masivo.

El conocimiento de las propiedades y características de implementaciones alternativas de un contenedor, tanto en memoria principal como secundaria.

Capacitar al alumno para seleccionar, diseñar e implementar las estructuras de datos adecuadas a una aplicación concreta.

Metodología

Durante las clases en el aula el profesor expondrá los contenidos de la asignatura. Estas clases serán teórico-prácticas, conjugándose las explicaciones teóricas con la exposición de ejemplos y realización de ejercicios por parte de los alumnos. En la presentación de la materia se empleará videoproector.

Las clases en laboratorio perseguirán que los alumnos adquieran destreza en el uso del lenguaje de programación C++ y habilidad en el empleo de los conceptos mostrados en clase. Los alumnos realizarán ejercicios prácticos guiados por el profesor.

Las tutorías serán un elemento básico de interacción profesor alumno. Tienen como objetivos resolver dudas, guiar a los alumnos en la ampliación de conocimientos y facilitarles bibliografía y orientación en caso necesario.

Como medio principal para suministrar información referente a la asignatura se empleará la página Web <http://www.gedlc.ulpgc.es>. La realización de las clases prácticas se soportará íntegramente vía Web. Se potenciará la interacción electrónica alumnos/profesor mediante correo electrónico.

Criterios de Evaluación

La evaluación de la asignatura consta de dos partes: una parte práctica y otra teórica. En la parte práctica se recogen las prácticas de laboratorio, un examen escrito práctico y un examen práctico de suficiencia. En la parte teórica se evaluarán los aspectos teóricos mediante un examen escrito.

* Examen escrito.

En el examen escrito el alumno deberá demostrar sus conocimientos teóricos y prácticos de la materia. Parte del examen escrito se corresponde con el “examen escrito práctico”, el resto con la

parte teórica. El examen estará dividido en ejercicios, cada uno de los cuales aportará un porcentaje especificado a la nota de cada parte.

*** Prácticas de laboratorio.**

La evaluación de las prácticas se efectuará a partir de los ejercicios prácticos que se irán planteando regularmente. El enunciado de cada ejercicio incluirá las especificaciones, plazos de entrega y porcentaje que aporta a la evaluación.

Por cada ejercicio práctico planteado el alumno deberá entregar, dentro de los plazos establecidos, los ficheros conteniendo el código fuente y demás información requerida, usando para la entrega el programa de gestión automática de prácticas (GAP), no admitiéndose ninguna otra vía de entrega.

El profesor someterá los programas e información entregados a los análisis y pruebas oportunos para formular una valoración global que tenga en cuenta todos los aspectos implicados en su realización, con especial énfasis en la ejecución (funcionamiento, adecuación a las especificaciones, robustez, ...) y en el estilo (formato, comentarios, elección de identificadores, ...). Cuando lo estime conveniente, el profesor podrá citar al alumno para formularle cuestiones que considere relevantes para la valoración global reseñada. Las prácticas en el laboratorio se valorarán de 0 a 10 puntos.

*** Examen práctico de suficiencia.**

Este examen consistirá en la realización de un ejercicio de programación en una situación real ante el ordenador. Su finalidad es valorar la efectividad de las habilidades alcanzadas en este ámbito. Se valorará con los mismos criterios que las prácticas pero se calificará como APTO o NO APTO.

CALIFICACIÓN

La valoración de la parte práctica consta, a partes iguales, de las prácticas de laboratorio y del “examen escrito práctico”. Además, para superar esta parte se requiere obtener la calificación de APTO en el examen práctico de suficiencia. De este último examen quedarán exonerados quienes obtengan resultados especialmente brillantes en el examen escrito práctico y/o en la valoración de las prácticas de laboratorio. No podrán ser exonerados de este examen los alumnos que no asistan al menos al 60% de las clases prácticas.

Para superar la asignatura habrá de superarse la parte práctica y la teórica, además de obtener una nota mínima de 5 puntos en la nota final. La nota final se calcula con la siguiente fórmula:

$$NF = NT * 0.6 + NP * 0.4$$

donde NF representa la nota final, NT la nota de la parte teórica, NP la nota de la parte práctica. La nota final de los alumnos que no cumplan las condiciones estipuladas para superar la asignatura será la obtenida en la parte teórica hasta un máximo de 4 puntos.

Los alumnos que incurran en fraude en alguna de las pruebas obtendrán una nota final de 0 puntos, independientemente de otras medidas que pudieran tomarse de acuerdo con lo establecido en los reglamentos, al respecto, de la ULPGC.

Descripción de las Prácticas

Las prácticas consistirán en el desarrollo de algoritmos para solucionar problemas simples y la implementación de pequeños programas o módulos de programas que utilicen los diferentes conceptos y elementos de programación presentados en las clases teóricas. En las prácticas en laboratorio, los programas se implementarán usando el lenguaje de programación C++ y deberán funcionar en el ordenador como requisito previo a cualquier evaluación.

Las prácticas incluirán tanto ejercicios de acción formativa como de acción sumativa que se señalarán adecuadamente en cada caso y para los que se establecerán plazos de entrega que deberán ser respetados para que sean tenidos en cuenta a efectos de evaluación.

Las prácticas de acción sumativa son:

Práctica número 1

Descripción: Clases, constructores y sobrecarga de operadores

Objetivos:

Adquirir destreza en la creación de clases y constructores. Saber aplicar la sobrecarga de operadores en C++

Nº horas estimadas en laboratorio: 3 horas

Práctica número 2

Descripción: Operador de asignación y constructor de copia

Objetivos:

Valorar la necesidad del constructor de copia y del operador de asignación en C++

Nº horas estimadas en laboratorio: 3 horas

Práctica número 3

Descripción: Herencia y polimorfismo

Objetivos:

Adquirir experiencia en el uso de herencia y polimorfismo en C++

Nº horas estimadas en laboratorio: 4 horas

Práctica número 4

Descripción: Parametrización

Objetivos:

Adquirir destreza en el uso de la genericidad en C++

Nº horas estimadas en laboratorio: 2 horas

Práctica número 5

Descripción: STL

Objetivos:

Adquirir experiencia en el uso de bibliotecas de clases.

Nº horas estimadas en laboratorio: 3 horas

Material de Laboratorio requerido.

Software: Entorno de desarrollo de C++ estándar que incluya editor, compilador y depurador.

Navegador de internet actualizado.

Hardware: Un ordenador por alumno que ejecute el software solicitado y que disponga de acceso a la red de la universidad.

Bibliografía

[1 Básico] Técnicas y metodología orientada a objetos. C++ /

Juan Carlos Rodríguez del Pino, Margarita Díaz Roca, José Daniel González Domínguez.

Universidad de Las Palmas de Gran Canaria,, Las Palmas de Gran Canaria : (1999)

84-699-0547-3

[2 Básico] Fundamentos de estructuras de datos: soluciones en Ada, Java y C++ /

Zenón José Hernández Figueroa, Juan Carlos Rodríguez del Pino, José Daniel González Domínguez, Margarita Díaz

Roca, José Rafael Pérez Aguilar, Gustavo Rodríguez Rodríguez ... [et al.].

Thomson,, Australia, España [etc.] : (2005)

84-9732-358-0

[3 Recomendado] Construcción de software orientado a objetos /

Bertrand Meyer.

Prentice Hall,, Madrid : (1998) - (2ª ed.)

84-8322-040-7

[4 Recomendado] El lenguaje de programación C++ /

Bjarne Stroustrup.

Addison Wesley,, Madrid : (2001) - (Edición especial.)

847829046X

[5 Recomendado] El lenguaje de programación C++ /

Bjarne Stroustrup.

Addison Wesley,, Argentina : (1998) - (3ª ed.)

84-7829-019-2

[6 Recomendado] Estructuras de datos, algoritmos y programación orientada a objetos /

Gregory L. Heileman ; traducción Joaquín Mateos Lago ; revision técnica Luis Hernández Yáñez.

, McGraw-Hill, Madrid, (1997)

8448111737

[7 Recomendado] Programación y diseño en C++: introducción a la programación y al diseño orientado a objetos /

James P. Cohoon, Jack W. Davidson.

McGraw-Hill,, Madrid : (2000) - (2ª ed.)

84-481-2682-3

[8 Recomendado] Estructuras de archivos: un conjunto de herramientas conceptuales /

Michael J. Folk, Bill Zoellick.

Addison-Wesley Iberoamericana,, Argentina : (1992)

0201629232

[9 Recomendado] File structures: an object - oriented approach with C++ /

Michael J. Folk, Bill Zoellick, Greg Riccardi.

Addison-Wesley,, Reading [etc.] : (1998)

0-201-87401-6

[10 Recomendado] Introduction to algorithms /

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest.

MIT,, Cambridge (Massachusetts) : (1990)

0-262-53091-0

Equipo Docente

OCTAVIO SANTANA SUÁREZ

Categoría: CATEDRÁTICO DE UNIVERSIDAD

Departamento: INFORMÁTICA Y SISTEMAS

Teléfono: 928458730 **Correo Electrónico:** octavio.santana@ulpgc.es

MARGARITA DÍAZ ROCA

Categoría: CATEDRÁTICO DE ESCUELA UNIVERSITARIA

Departamento: INFORMÁTICA Y SISTEMAS

Teléfono: 928458732 **Correo Electrónico:** margarita.diaz@ulpgc.es

WEB Personal: <http://www2.dis.ulpgc.es/~mdiaz/>

JUAN CARLOS RODRÍGUEZ DEL PINO

(COORDINADOR)

Categoría: TITULAR DE ESCUELA UNIVERSITARIA

Departamento: INFORMÁTICA Y SISTEMAS

Teléfono: 928458733 **Correo Electrónico:** jc.rodriguezdelpino@ulpgc.es

JOSÉ DANIEL GONZÁLEZ DOMÍNGUEZ

(RESPONSABLE DE PRACTICAS)

Categoría: TITULAR DE ESCUELA UNIVERSITARIA

Departamento: INFORMÁTICA Y SISTEMAS

Teléfono: 928458751 **Correo Electrónico:** josedaniel.gonzalez@ulpgc.es