



UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

GUÍA DOCENTE

CURSO: 2011/12

12537 - ESTRUCTURAS DE DATOS II

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], [CO90]

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], [CO90]

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 de un lenguaje de programación orientado a objetos.

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 emplearán entre otros medios un videoprojector y un ordenador.

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

Las tutorías complementarán el seguimiento personalizado de los alumnos. Se potenciará la interacción electrónica entre los alumnos y el profesor.

Criterios de Evaluación

La evaluación de la asignatura tendrá en cuenta tanto los aspectos teóricos como los prácticos.

En la medida de las posibilidades se procurará una evaluación continuada y personalizada. Con la finalidad de llevar a cabo una evaluación continuada y personalizada de la manera más idónea, se seleccionarán los mecanismos más adecuados en función de las características de cada grupo de entre los múltiples recomendados en la amplia bibliografía sobre evaluación continua. En esta evaluación se atenderán aspectos como la asistencia y la participación activa en las sesiones, las pruebas teórico-prácticas y los trabajos individuales o colectivos que se puedan realizar.

Se dispondrá de otros dos mecanismos: un examen escrito y la evaluación del trabajo de laboratorio.

*** Examen escrito.**

En el examen escrito el alumno deberá demostrar sus conocimientos teóricos y prácticos. El examen estará dividido en ejercicios, cada uno de los cuales aportará un porcentaje especificado a la nota. De este examen podrán quedar exonerados aquellos alumnos cuya evaluación personalizada así lo aconseje. Se valorará de 0 a 10 puntos.

*** Evaluación del trabajo práctico.**

La evaluación del trabajo práctico se valorará de 0 a 10 puntos y se efectuará a partir de los ejercicios de laboratorio que se irán planteando regularmente o del trabajo alternativo.

El enunciado de cada ejercicio de laboratorio incluirá las especificaciones, plazos de entrega y porcentaje que aporta a la evaluación del trabajo práctico, así como su desarrollo individual o en grupo.

Por cada ejercicio de laboratorio planteado el alumno deberá entregar, dentro de los plazos establecidos, los ficheros conteniendo el código fuente y demás información requerida. El profesor tendrá en cuenta todos los aspectos implicados: realización, ejecución (funcionamiento, adecuación a las especificaciones, robustez,...) y estilo (formato, comentarios, elección de identificadores,...). Cuando lo estime conveniente, el profesor podrá citar al alumno para formularle cuestiones que considere relevantes para su valoración.

De manera alternativa, y cuando las posibilidades lo permitan, podrá consistir en el desarrollo de un trabajo singular sobre estructuras de datos.

CALIFICACIÓN

Para superar la asignatura habrá de obtenerse una nota mínima de 5 puntos según la siguiente fórmula:

$$NF = ECP * 0.7 + ETP * 0.3$$

donde NF representa la nota final, ECP la evaluación continua personalizada y ETP la evaluación del trabajo práctico.

Aquellos alumnos que no superen la asignatura podrán hacer uso del mecanismo alternativo proporcionado por el examen escrito.

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 presentados en las clases teóricas. En las prácticas en laboratorio, los programas se implementarán usando un lenguaje de programación orientado a objetos y deberán funcionar en el entorno asignado a las prácticas como requisito previo a cualquier evaluación.

Las prácticas incluirán ejercicios para los que se establecerán plazos de entrega que deberán ser respetados para que sean tenidos en cuenta a efectos de evaluación.

El listado de prácticas previstas es:

Práctica número 1

Descripción: Conceptos de programación orientada a objetos I: Clases, constructores, sobrecarga de operadores, operador de asignación y constructor de copia.

Objetivos:

Adquirir destreza en la creación de clases, constructores, sobrecarga de operadores, constructor de copia y operador de asignación.

Nº horas estimadas en laboratorio: 3 horas

Práctica número 2

Descripción: Conceptos de programación orientada a objetos II: Herencia, polimorfismo y parametrización.

Objetivos:

Adquirir destreza en el uso de la herencia, del polimorfismo y de la genericidad.

Nº horas estimadas en laboratorio: 3 horas

Práctica número 3

Descripción: Métodos sobre una estructura TRIE.

Objetivos:

Adquirir destreza en el manejo del TRIE.

Nº horas estimadas en laboratorio: 4 horas

Práctica número 4

Descripción: Métodos sobre una estructura multirrama.

Objetivos:

Adquirir destreza en el manejo de una estructura multirrama.

Nº horas estimadas en laboratorio: 5 horas

Material de Laboratorio requerido.

Software: Entorno de desarrollo que incluya editor, compilador y depurador. Navegador de Internet.

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

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*