
Fundamentos de computadores y lenguajes

Programa de la asignatura

Página Web: <http://www.ctr.unican.es/asignaturas/fundamentos>

Curso 2008-2009

Profesores

Michael González Harbour (mgh@unican.es)

José Ignacio Espeso (espesoji@unican.es)

José Carlos Palencia Gutiérrez (palencij@unican.es)

Asignatura	Curso	Cuatr.	Créditos
Fundamentos de computadores y lenguajes	1	1-2	6+3
Programación de alto nivel	2	2	3+3
Sistemas operativos	3	2	3+3
Programación en entorno científico	3	2	3+3
Ingeniería de programación	4	1	3+3
Sistemas informáticos	5	2	3+3
Instrumentación electrónica	5	2	3+3

Objetivos de "fundamentos de computadores y lenguajes"

- Adquirir conocimientos básicos del computador
- Conocer las características generales de los principales lenguajes de programación
- Conocer los principios de la modularidad y abstracción para crear módulos de programa sencillos
- Conocer los pasos del ciclo de vida de los programas
- Adquirir conocimientos básicos de programación orientada a objetos
- Conocer los principios de la entrada/salida: interactiva y con ficheros
- Conocer un sistema operativo a nivel de usuario
- Conocer los tipos de datos sencillos

Competencias

- Utilizar un sistema operativo a nivel de usuario
- Saber diseñar, programar y probar pequeños algoritmos
- Saber utilizar módulos de programa para hacer un programa más complejo
- Crear módulos de programa, separando las fases de diseño e implementación
- Implementar programas sencillos que sean fiables y fáciles de entender
- Aplicar estrategias de prueba sencillas para un módulo de programa
- Saber documentar un proyecto de programación

2. Programa de la asignatura

Bloque I: El computador y el proceso de programación

Bloque II: Lenguaje de Programación

Bloque III: Herramientas

Bloque I: El computador y el proceso de programación



1. Introducción a los computadores y su programación

- Arquitectura básica de un computador. El software del sistema. Lenguajes de alto nivel. El proceso de compilación. El ciclo de vida del software.

2. Introducción al análisis y diseño de algoritmos.

- Diseño de un programa. Concepto de algoritmo. Descripción de algoritmos: el pseudolenguaje. Tiempo de ejecución de algoritmos. La notación $O(n)$. Ejemplos.

3. Introducción al análisis y diseño de programas

- Actividades del ciclo de vida del software. Paradigmas de desarrollo de programas. Análisis y especificación. Diseño arquitectónico. Técnicas de diseño detallado.

4. Verificación de programas

- Importancia de la verificación. Estrategias de prueba. Fases de la prueba de programas. Depuración. Elección de datos para la prueba.

Bloque II: Lenguaje de Programación



1. Elementos básicos de los lenguajes

- Introducción. Concepto de clase y objeto. Estructura de un programa. Tipos primitivos. Variables y constantes. Operadores y expresiones. Conversión de tipos. Uso de funciones matemáticas. Declaración de clases y objetos. Strings. Composición de objetos. Atributos y métodos estáticos.

2. Instrucciones de control.

- Instrucción condicional. Instrucción condicional múltiple. Instrucciones de lazo.

3. Tipos de datos.

- Tipos enteros. Tipos reales. Conversión de tipos. Caracteres. Strings. Arrays. Tipos enumerados.

4. Métodos

- Conceptos básicos. Paso de parámetros. Constructores. Reglas de visibilidad.

Bloque II: Lenguaje de Programación (cont.)



5. Modularidad y Abstracción

- Paquetes. Especificaciones de acceso. Abstracción mediante componentes privados. Documentación. Tipos de módulos de programa.

6. Tratamiento de errores.

- Transmisión de errores mediante parámetros. Excepciones. Errores o excepciones predefinidos. Uso de excepciones. La cláusula *finally*. Principales métodos de gestión de los errores. Cuándo usar excepciones.

7. Entrada/salida simple.

- Entrada/salida de texto y de caracteres. E/S de números. Operaciones para gestión de pantalla y teclado. Gráficas. Dibujos. Menús de botones.

8. Entrada/salida avanzada

- Ficheros. E/S de objetos y variables. E/S de texto. Salida de texto formateado.

Bloque III: Herramientas



1. Uso de sistemas operativos

- Conceptos básicos. Sistemas operativos comunes. Arranque y finalización de sesiones. El sistema de ficheros. El intérprete de órdenes. Ejecución de programas. El gestor gráfico de ficheros. Uso de la memoria USB. Aspectos avanzados.

2. Uso de un entorno integrado de desarrollo de programas

- Proceso de desarrollo de programas. Editor de texto. Editor de texto orientado al lenguaje. El compilador y la ejecución. Entorno integrado de desarrollo. La depuración. Generación de documentos. Empaquetamiento del programa

Otras actividades (Seminarios)

- Creación de páginas Web.
- Creación de *applets* sencillos.

Distribución de las clases de teoría, problemas y prácticas

Primer cuatrimestre:

- 1 hora teoría + 1 hora prácticas

Hora	Lunes	Martes	Miércoles	Jueves	Viernes
12:45-13:45	Teoría Aula 4	Prácticas Grupo 1 LSC1	Prácticas Grupo 2 LSC1		

- 1 hora semanal de trabajo de preparación/finalización de problemas, y ejercicios propuestos
- 2 horas semanales de actividades de estudio (algo más al final del cuatrimestre): ejercicios

Distribución de las clases, segundo cuatrimestre

- 2 horas teoría + 2 horas prácticas

Hora	Lunes	Martes	Miércoles	Jueves	Viernes
10:45-11:45	Prácticas Grupo 1 LSC1		Prácticas Grupo 2 LSC1		
11:45-12:45		Teoría Aula 4		Teoría Aula 4	

- 1 hora semanal de trabajo de preparación de prácticas
- 3.5 horas semanales de estudio: ejercicios

Prácticas, en número a determinar:

- utilización del sistema operativo
- realización de programas simples para adquirir soltura con el lenguaje de programación
- uso de un módulo de programa ya desarrollado, para construir programas más complejos.
- desarrollo de módulos de programa
- verificación de programas

Proceso para evaluar prácticas

- presentación en el laboratorio y visto bueno del profesor
- breve memoria, según formato establecido

6. Sistemas de evaluación

Dos exámenes parciales de cuestiones y problemas (feb, jun)

Dos exámenes finales de cuestiones y problemas (jun, sep)

Se pueden usar apuntes y libros

Evaluación de las prácticas:

- continuada en el laboratorio
- examen de prácticas y entrega de todas las prácticas, en casos justificados

Nota Final:

- 60% examen, 40% prácticas
- Es preciso superar tanto el examen como las prácticas

Criterios de evaluación de las prácticas



a) Trabajo en el laboratorio

- **Conocimientos:** conocimiento de los conceptos de estructuras de datos y algoritmos
- **Grado de Resolución:** grado de resolución de la práctica alcanzado durante las horas en el laboratorio

b) Informes sobre las prácticas

- **Estilo:** estilo de programación, claridad del código, documentación del código
- **Informe:** organización y claridad en la exposición
- **Fecha de entrega**

Criterios de evaluación de las prácticas (cont.)



Fecha de entrega

- **Objetivo:** conseguir una evaluación realmente continuada.

Las prácticas cuyo informe se presente con retraso:

- de hasta una semana: un punto menos (sobre 10)
- entre una y dos semanas: tres puntos menos

Las entregas pasadas dos semanas del plazo no se evaluarán, pero su presentación es obligatoria

Normas para la evaluación

La realización de prácticas y exámenes y su evaluación es individual

Reglamento de exámenes:

- cualquier alumno que se atribuya indebidamente la autoría de trabajos académicos requeridos para la evaluación, tendrá la calificación de "suspense" o de "0"

El objetivo de las prácticas no es hacerlas

- es aprender
- y evaluar los conocimientos y habilidades

7. Bibliografía: Libros recomendados

- [1] D.J. Barnes y M. Kölling, "Objects first with Java: A practical introduction using Bluej", Prentice Hall, 2003.
- [2] Ken Arnold, James Gosling, David Holmes, "El lenguaje de programación Java", 3ª edición. Addison-Wesley, 2000, ISBN 84-7829-045-1
- [3] Patrick Naughton, Herbert Schildt, Herb Schildt. "Java 2, The complete reference". Osborne McGraw-Hill 1999. ISBN: 0072119764
- [4] David M. Arnow, Gerald Weiss. "Introducción a la programación con Java: Un enfoque orientado a objetos". Prentice Hall, 2000. ISBN 84-7829-033-8
- [5] E. Bueno, y otros. "Algoritmos y Ejercicios resueltos en Java". Prentice Hall 2003. ISBN 84-705-4024-2

Bibliografía: Libros de consulta:



- [6] B. Eckel, “Piensa en Java”, Addison Wesley, 1002. ISBN 84-205-3192-8
- [7] H.M. Deitel y P.J. Deitel “Cómo programar en Java”. Prentice Hall, 1997. ISBN 970-17-0044-9
- [8] Duane A. Bailey, Duane W. Bailey, “Java Elements: Principles of Programming in Java” McGraw Hill, ISBN 0072285372
- [9] Judy Bishop. “Java. Fundamentos de programación” 2ª edición, Addison Wesley 1999. ISBN 84-7829-022-2
- [10] Judy Bishop. “Java Gently”. 3rd. edition, Addison Wesley, 2001
- [11] John Hunt and P.J. Thomas. “Java and Object-Orientation: An Introduction (Applied Computing)”. Springer Verlag, 1998.
- [12] Walter Savitch, Walter Szvitch, and Richard Johnsonbaugh. “Java: An Introduction to Computer Science and Programming”. Prentice Hall, 1998. ISBN 0132874261.

Bibliografía: Libros de consulta (cont):



- [13] Dennis Kafura, “Object-Oriented Software Design and Construction With Java”, Prentice Hall, 2000, ISBN: 013011264X
- [14] Steve Holzner, “La biblia de Java 2”, Anaya, 2000, ISBN 84-415-1037-7
- [15] K.N. King, “Java Programming: From the Beginning”, W.W. Norton & Co, 2000, ISBN 0393974375
- [16] Herbert Schildt, “Java 2: A Beginner’s Guide”, McGrawHill, 2000, ISBN 0072127422
- [17] Aho y Ullman, “Foundations of Computer Science”. W H Freeman & Co. 1995, ISBN: 0716782847
- [18] J. Castro, F. Cucker, X. Messeger, A. Rubio L. Solano, B. Valles. “Curso de Programacion”. McGraw- Hill, 1993.