



Universidad
Nebrija



**GRADO EN
INGENIERÍA
INFORMÁTICA
CURSO 2008/2009**

**Asignatura: Metodología y
Técnicas de Programación II**
Código: IIN108



Asignatura: IIN108 Metodología y Técnicas de Programación II

Formación: Básica

Créditos ECTS: 6

Curso: Primero

Semestre: Segundo

Profesor: Luis Izquierdo Mesa

Curso académico: --

1. REQUISITOS PREVIOS

Haber aprobado la asignatura Metodología y Técnicas de Programación I (IIN107). Será necesario consultar bibliografía y analizar textos en inglés, por lo que es preciso tener conocimientos de este idioma.

2. BREVE DESCRIPCIÓN DE CONTENIDOS

Se tratarán los aspectos fundamentales de la programación orientada a objetos, básicos para entender otros paradigmas de programación y necesarios para posteriores asignaturas.

Se tratarán los fundamentos teóricos de la programación, con ejemplos concretos, utilizando el lenguaje de programación C++.

3. COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE Y RESULTADOS DEL APRENDIZAJE

Los estudiantes habrán demostrado poseer y comprender los conocimientos relativos a la programación orientada a objetos, entendida ésta como un conjunto de estructuras modulares basadas en datos, que se agrupan en clases que incluyen los procedimientos de acceso y comportamiento de esos datos, servicios o métodos de la clase.

Los estudiantes habrán adquirido la destreza necesaria para el manejo de ciertas técnicas que son básicas para el óptimo uso de esta metodología:

- Diseño de clases.
- Herencia
- Polimorfismo
- Sobrecargas
- Reusabilidad mediante plantillas.

Todas estas técnicas ayudarán al alumno a estrechar la franja que separa el mundo real del entorno informático, de tal manera que el modelo que se construya se parezca lo máximo posible al problema planteado originalmente.

Los estudiantes habrán desarrollado habilidades de aprendizaje que les permitan emprender las asignaturas posteriores de “Metodología y Técnicas de Programación III”, con un alto grado de autonomía.

4. ACTIVIDADES FORMATIVAS Y METODOLOGÍA

Clases de teoría y problemas: (1,8 créditos ECTS) Clases en aula de ordenadores donde se combinará la lección magistral con el trabajo en el ordenador por parte del profesor y de los alumnos. Se explicarán los fundamentos teóricos de la programación orientada a objetos necesarios para el diseño y reutilización, por medio de las técnicas explicadas anteriormente, de los objetos.

Clases prácticas en laboratorio: (0,6 créditos ECTS) Prácticas en ordenador supervisadas por el profesor.

Se implementarán en ordenador sistemas informáticos que resuelvan distintos problemas utilizando el lenguaje de programación C++. El alumno entregará un informe personalizado con los resultados obtenidos en estas prácticas.

Estudio individual: (3,2 créditos ECTS) Trabajo individual del alumno.

Tutorías: (0,4 créditos ECTS) Clases para resolución de cuestiones.

5. SISTEMA DE EVALUACIÓN

5.1. Convocatoria Ordinaria:

Las prácticas supondrán un 25% de la nota, mientras que el examen parcial y final constituirán un 15 y un 60%, respectivamente.

5.2. Convocatoria Extraordinaria:

Las prácticas supondrán un 20% de la nota, mientras que el examen final constituirá un 70%.

5.3. Restricciones:

Para poder hacer la suma ponderada de las calificaciones anteriores, es necesario: la asistencia a las clases como mínimo del 80 % de las horas presenciales, y obtener al menos un cinco en el examen final correspondiente. El alumno con nota inferior se considerara suspenso.

Es imprescindible la entrega de las prácticas, tanto para la convocatoria ordinaria, como para la extraordinaria.

Es imprescindible la asistencia a las sesiones de laboratorio, considerándose como suspendida la asignatura en su convocatoria ordinaria en el caso de la falta de asistencia injustificada a más de un 25% de estas sesiones.

El examen parcial no libera material y es obligatorio.



No está permitido el uso de calculadoras en los exámenes ni de ningún otro tipo de material que no sea el proporcionado por el profesor.

6. BIBLIOGRAFÍA

- **Bibliografía Básica**

- Bruce Eckel. "Thinking in C++ 2e". Prentice Hall, 2000. Versión descargable en Inglés: <http://mindview.net/Books/TICPP/ThinkingInCPP2e.html>.
- Deitel, H.M. y Deitel P. J. "C++. Cómo programar". Prentice Hall. México 1999.

- **Bibliografía Complementaria**

- Bjarne Stroustrup. "El Lenguaje de Programación C++". Edición especial. Addison Wesley, 2002.
- Bjarne Stroustrup. "The Design and Evolution of C++". Edición especial. Addison Wesley, 1995.
- Scott Meyers. Effective C++ 2e: "50 Specific Ways to improve your programs and designs". Addison Wesley, 1997.

7. LOCALIZACIÓN DEL PROFESOR

Prof. Luis Izquierdo Mesa
Departamento de Ingeniería Informática
Aula 205
E-Mail: lizquier@nebrija.es
Tfno: +34 - 91.452.11.00 – Extensión 2864

9. CONTENIDO DETALLADO DE LA ASIGNATURA

GRADO: Ingeniería Informática - **ASIGNATURA:** Metodología y Técnicas de Programación II - **CURSO:** Segundo
SEMESTRE: Primero - **CRÉDITOS ECTS:** 6

Sesión	Sesiones de Teoría, Problemas y Evaluación continua	Estudio individual y trabajos del alumno	Horas Presenciales	Horas Estudio y Trabajo
1	Introducción a la POO, organización de ficheros y notación húngara		1,5	
2	Asignación dinámica de la memoria. Operadores new y delete	Hoja ejercicios 1	1,5	6
3	Asignación de memoria dinámica para arrays. Arrays dinámicos		1,5	
4	Diferencias entre un array estático y un array dinámico		1,5	
5	Programación OO. Clases y objetos		1,5	
6	Encapsular, acceso a miembros de una clase	Hoja Ejercicios 2	1,5	7
7	Atributos y métodos		1,5	
8	Constructores y destructores		1,5	
9	Excepciones, Lanzamiento y captura de excepciones		1,5	
10	Punteros a objetos, referencias a objetos		1,5	
11	Clases friends		1,5	
12	CORRECCION DEL EXAMEN PARCIAL		1,5	
13	Funciones friends	1,5		
14	EXAMEN PARCIAL		1,5	7
15	CORRECCIÓN DEL EXAMEN PARCIAL	Hoja Ejercicios 3	1,5	7
16	Sobrecarga de operadores unitarios y binarios		1,5	
17	Sobrecarga de operadores lógicos y relacionales		1,5	
18	Sobrecarga de operadores de inserción y extracción	Hoja Ejercicios 4	1,5	7
19	Constructor de copia, operator=, destructor		1,5	
20	Templates. Plantillas de funciones.		1,5	
21	Templates II. Plantillas de clases		1,5	
22	Herencia	Hoja ejercicios 5	1,5	7
23	Herencia múltiple y conversión de clases		1,5	
24	Accesibilidad a las clases derivadas		1,5	
25	Constructores y destructores en la herencia		1,5	
26	Clases base virtuales	Hoja Ejercicios 6	1,5	7
27	Punteros a clases derivadas		1,5	
28	Funciones virtuales puras y clases abstractas		1,5	
29	Polimorfismo		1,5	
	Evaluación Final Ordinaria Extarordinaria		1,5	12
	Tutorías		10,0	
1	Sesión 1	Sesiones de laboratorio	1,5	2
2	Sesión 2	Sesiones de laboratorio	1,5	2
3	Sesión 3	Sesiones de laboratorio	1,5	2
4	Sesión 4	Sesiones de	1,5	2



		laboratorio		
5	Sesión 5	Sesiones de laboratorio	1,5	2
6	Sesión 6	Sesiones de laboratorio	1,5	2
7	Sesión 7	Sesiones de laboratorio	1,5	2
8	Sesión 8	Sesiones de laboratorio	1,5	2
9	Sesión 9	Sesiones de laboratorio	1,5	2
10	Sesión 10	Sesiones de laboratorio	1,5	2
		TOTAL:	70,0	80,0
		TOTAL HORAS:		150,0