DESCRIPCIÓN DE LA ASIGNATURA

Grado/Máster en:	Graduado/a en Ingeniería del Software por la Universidad de Málaga
Centro:	Escuela Técnica Superior de Ingeniería Informática
Asignatura:	Programación Orientada a Objetos
Código:	109
Tipo:	Formación básica
Materia:	Informática
Módulo:	Formación básica
Experimentalidad:	69 % teórica y 31 % práctica
Idioma en el que se imparte:	Inglés, Español
Curso:	1
Semestre:	2
Nº Créditos	6
Nº Horas de dedicación del estudiante:	150
Nº Horas presenciales:	60
Tamaño del Grupo Grande:	72
Tamaño del Grupo Reducido:	30
Página web de la asignatura:	

EQUIPO DOCENTE

Departamento: LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

Área: LENGUAJES Y SISTEMAS INFORMÁTICOS

Nombre y Apellidos	Mail	Teléfono Laboral	Despacho	Horario Tutorías
Coordinador/a: FRANCISCO GUTIERREZ LOPEZ	fgutierrez@uma.es	952133314	3.2.46 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Lunes 10:45 - 12:45, Miércoles 10:45 - 12:45, Martes 10:45 - 12:45
FRANCISCO JAVIER DURAN MUÑOZ	fdm@uma.es	952132820	3.2.46 - E.T.S.I. INFORMÁTICA	
PABLO LOPEZ OLIVAS	plopez@uma.es	952133305	3.2.50 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Lunes 12:30 - 14:00, Martes 12:00 - 14:00, Jueves 10:00 - 12:30
VICENTE JESUS BENJUMEA GARCIA	vjbenjumea@uma.es	952132754	3.2.3 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Lunes 16:30 - 18:30, Viernes 10:45 - 12:45, Miércoles 12:30 - 14:30 Segundo cuatrimestre: Lunes 15:15 - 17:15, Viernes 15:15 - 17:15, Miércoles 15:15 - 17:15
BARTOLOME RUBIO MUÑOZ	brubio@uma.es	952132753	3.2.45 - E.T.S.I. INFORMÁTICA	Todo el curso: Viernes 09:00 - 10:30, Martes 09:00 - 10:30, Miércoles 09:00 - 10:30, Lunes 09:00 - 10:30
MONICA TRELLA LOPEZ	mtl@uma.es	952137152	3.2.34 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Lunes 08:30 - 10:30, Jueves 15:00 - 17:00, Jueves 08:30 - 09:30, Lunes 13:00 - 14:00

RECOMENDACIONES Y ORIENTACIONES

Se espera que el alumno haya asimilado los conocimientos impartidos en la asignatura Fundamentos de Programación del primer cuatrimestre.

==========

Students are expected to have the knowledge imparted in the subject Programming Fundamentals during the first semester.

CONTEXTO

Esta asignatura forma parte del módulo de formación básica en la materia Informática.

Se sitúa en el segundo cuatrimestre del primer curso cuando el alumno ya tiene conocimientos previos de programación imperativa y servirá de base para el resto de asignaturas relacionadas con el desarrollo de software orientado a objetos.

==========

This subject is part of the basic formation module in the field of Computer Science.

It is imparted in the second semester of the first year, when the student already has previous knowledge about imperative programming and will be the basis for the rest of the subjects related to the development of object-oriented software.

COMPETENCIAS

1 Competencias generales y básicas.

BÁSICAS

CB01 Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

CB02 Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

CB04 Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB05 Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

GENERALES

CG08 Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

CG09 Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática

CG10 Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias básicas, comunes y específicas del título.

2 Competencias específicas.

Formacion Basica

CE-CB03 Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.

CE-CB04 Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

CE-CB05 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.

CONTENIDOS DE LA ASIGNATURA

Tema 1: Introducción a la POO / Unit 1. Introduction to Object-Oriented Programming

1.1 Evolución de los lenguajes de programación

1.2 Conceptos fundamentales de la POO

==========

1.1 Evolution of Programming Languages

1.2 Main concepts of the OOP paradigm

Tema 2: Introducción a Java / Unit 2. Introduction to Java

2.1 Introducción histórica

2.2 Programas y paquetes

2.3 Clases y objetos

2.4 Elementos del lenguaje

2.5 Control de errores

2.6 Cadenas de caracteres

2.7 Arrays

2.8 Herencia

2.9 Clases abstractas e interfaces

==========

2.1 Historical Introduction
2.2 Programs and Packages
2.3 Classes and Objects
2.4 Language Elements
2.5 Error Control
2.6 Strings
2.7 Arrays
2.8 Inheritance
2.9 Abstract Classes and Interfaces

Tema 3: Tratamiento de errores, excepciones / Unit 3. Exceptions

3.1 Software tolerante a fallos. El concepto de excepción
3.2 Captura y tratamiento de excepciones
3.3 Propagación de excepciones
3.4 Excepciones predefinidas
3.5 Definición de nuevas excepciones

==========

3.1 Fault Tolerance Software
3.2 The concept of Exception
3.3 Catching and Managing Exceptions
3.4 Exception Propagation
3.5 Predefined Exception
3.6 User-defined Exception

Tema 4: Clases básicas predefinidas. Entrada/Salida / Unit 4. Predefined Java Classes. Input/Output

4.1 Organización en paquetes
4.2 Clases básicas del paquete java.lang
4.3 Clases básicas del paquete java.util
4.4 Clases básicas del paquete java.io

==========

4.1 Package organization
4.2 Basic Clases: java.lang package
4.3 Basic Clases: java.util package
4.4 Input/Output: java.io package

Tema 5: Colecciones de objetos / Unit 5. Collections and Iterators

5.1 Genericidad
5.2 Las interfaces de colecciones
5.3 Pilas, colas, listas, conjuntos y correspondencias
5.4 Iteradores
5.5 Ordenación de clases

5.6 Colecciones ordenadas

==========

5.1 Generic classes
5.2 Basic Interfaces and implementations
5.3 Sets, Lists and Maps
5.4 Iterators
5.5 Ordered Collections
5.6 Ordered Sets and Maps

Tema 6: Interfaces Gráficas de Usuario / Unit 6. Graphical User Interfaces

6.1 El patrón de diseño MVC
6.2 Componentes y contenedores
6.3 Gestores de esquemas
6.4 El modelo de eventos

==========

6.1 The MVC pattern
6.2 The Model
6.3 The View
6.4 The Controller

ACTIVIDADES FORMATIVAS

Actividades presenciales

Actividades expositivas
Lección magistral

Actividades prácticas en instalaciones específicas
Prácticas en laboratorio

ACTIVIDADES DE EVALUACIÓN

Actividades de evaluación presenciales

Actividades de evaluación del estudiante
Examen parcial
Examen final
Realización de trabajos y/o proyectos

RESULTADOS DE APRENDIZAJE / CRITERIOS DE EVALUACIÓN

RA1. Plantear y diseñar soluciones algorítmicas a problemas concretos mediante el uso de la programación orientada a objetos.
RA2. Valorar la importancia de la abstracción, especialmente con respecto a la programación de sistemas de cierta envergadura.
RA3. Estructurar el código usando los tipos de datos simples y estructurados y las estructuras de selección e iteración y la recursividad.
RA4. Diseñar programas aplicando los conceptos de la programación orientada a objetos: encapsulación, abstracción, herencia, polimorfismo y vinculación dinámica.
RA5. Diseñar, implementar y utilizar componentes de software reutilizables.
RA5. Desarrollar programas robustos, y tratar las excepciones producidas durante la ejecución de un programa.
RA6. Escribir programas para el manejo de eventos simples que respondan a la interacción con el usuario.
RA7. Evaluar de forma básica la complejidad y la corrección de algoritmos simples y la correcta elección de las estructura de datos.

RA8. Utilizar entornos y herramientas de desarrollo con los que implementar los algoritmos diseñados con un lenguaje de programación concreto.
RA9. Identificar, localizar y corregir los errores que puedan aparecer en las soluciones obtenidas para los problemas planteados.
RA10. Implementar, probar y depurar programas en un lenguaje orientado a objetos.

Estos RA desarrollan claramente las Competencias Específicas fijadas. En concreto, se desarrolla:

- La capacidad para comprender y dominar los conceptos básicos de algorítmica y su aplicación para la resolución de problemas (CE-CB03)
- Los conocimientos básicos sobre programación de los ordenadores (CE-CB04)
- El conocimiento de la estructura, organización y funcionamiento de los sistemas informáticos, los fundamentos de su programación y su aplicación para la resolución de problemas (CE-CB05)

Por otro lado, las Competencias Generales establecidas para la asignatura quedan también cubiertas con estos RA. Concretamente:
- La competencia CG08 está cubierta fundamentalmente por RA1, RA2 y RA8.
- La competencia CG09 queda cubierta por RA1, RA3 y RA5.
- La competencia CG10 queda cubierta por RA1, RA4, RA6, RA7, RA9 y RA10.

Tanto para la realización de las prácticas de laboratorio como la resolución de los problemas propuestos a lo largo de la asignatura los estudiantes aplicarán sus capacidades y conocimientos adquiridos conforme a lo reflejado en las Competencias Básicas establecidas (CB01, CB02, CB04 y CB05).

==========

These learning results clearly develop the course competencies. Specifically, it develops:

- The ability to understand the basic concepts of algorithm and its application for solving problems (CE-CB03)
- Basic knowledge about computer programming (CE-CB04)
- Knowledge of the structure, organization and operation of computer systems, the foundations of their programming and their application for solving problems (CE-CB05)

On the other hand, the general competences established for the subject are also covered with these learning results. Specifically:
- The CG08 competence is fundamentally covered by LR1, LR2 and LR8.
- The CG09 competence is covered by LR1, LR3 and LR5.
- The CG10 competence is covered by LR1, LR4, LR6, LR7, LR9 and LR10.

Both for the laboratory practices and the resolution of the problems proposed throughout the course students will apply their skills and knowledge as reflected in the basic competencies (CB01, CB02, CB04 and CB05).

On the other hand, the evaluation activities (midterm exams and final exam, with a theoretical component and another practical one), together with the evaluation procedure, clearly serve to evaluate if the students are able to reach the learning results.

Learning Results / Evaluation criteria
LR1. Propose and design algorithmic solutions to specific problems through the use of object-oriented programming.
LR2. Evaluate the importance of abstraction, especially with respect to the programming of systems of a certain size.
LR3. Structure the code using simple and structured data types and selection and iteration structures and recursion.
LR4. Designing programs applying the concepts of object-oriented programming: encapsulation, abstraction, inheritance, polymorphism and dynamic linking.
LR5. Design, implement and use reusable software components.
LR6. Develop robust programs, and manage the exceptions produced during the execution of a program.
LR7. Write programs for the management of simple events that respond to the interaction with the user.
LR8. Evaluate the complexity and correctness of simple algorithms and the correct choice of data structures.
LR9. Use environments and development tools with which to implement the algorithms designed with a specific programming language.

PROCEDIMIENTO DE EVALUACIÓN

Los alumnos entregarán todas las prácticas que se realicen a lo largo del curso. Además, se realizarán pequeños controles sobre las materias que se están desarrollando.
También realizarán un control programado obligatorio.

Los alumnos que hayan entregado un mínimo del 60% de las actividades propuestas y realicen el control obligatorio se considerará que siguen el proceso de evaluación continua.

Para los alumnos que siguen el proceso de evaluación continua, la calificación en las convocatorias de Junio y Septiembre se realiza de la siguiente forma:
- Se califican las actividades realizadas durante el curso sobre un máximo de 3 puntos.

- Se califica un examen final con un máximo de 10 puntos menos lo obtenido en el apartado anterior.

Para calcular la calificación en las convocatorias de Junio y Septiembre de los alumnos que no han seguido el proceso de evaluación continua y para todos los alumnos el resto de las convocatorias extraordinarias:
- Se califica un examen final sobre un máximo de 7 puntos.
- Se califica un cuestionario teórico sobre un máximo de 3 puntos.

En todos los casos, se suman las dos calificaciones y la calificación mínima para aprobar es de 5 puntos.

Para las convocatorias extraordinarias solo existirá un único examen final que puntúa el 100% de la calificación.

El profesor se reserva el derecho de verificar la calificación obtenida por el alumno, tanto en los controles como en los exámenes finales, realizando la correspondiente revisión en presencia del mismo.

Los alumnos a tiempo parcial y los alumnos deportistas universitarios seguirán el mismo proceso que los alumnos que no siguen el proceso de evaluación continua.

El plagio de cualquier actividad de la asignatura supondrá una calificación de 0 puntos para los implicados en la convocatoria correspondiente.

==========

The students will submit all the practices that are implemented throughout the course. In addition, there will be small tests on the contents being developed.
They will also perform a mandatory midterm exam.

Students who have submitted a minimum of 60% of the proposed activities and perform compulsory midterm exam will be considered to follow the continuous evaluation.

For students who follow the continuous evaluation, the qualification in the June and September exams is done as follows:
- The activities carried out during the course are rated on a maximum of 3 points.
- A final exam is qualified with a maximum of 10 points less the mark obtained in the previous section.

To calculate the mark in the June and September exams for students who have not followed the continuous evaluation and for all students the rest of the extraordinary examination calls:
- A final exam is rated on a maximum of 7 points.
- A theoretical questionnaire is rated on a maximum of 3 points.

In all cases, the two marks are added together and the minimum qualification to pass is 5 points.

For extraordinary examination calls there will only be one final exam that scores 100% of the grade.

The teacher reserves the right to verify the grade obtained by the student, both in the midterm and in the final exams, making the corresponding revision in the presence of the students.

Part-time students and university athletes will follow the same process as students who do not follow the continuous evaluation.

Committing plagiarism in a course task implies failing the course in that call for all the people involved.

BIBLIOGRAFÍA Y OTROS RECURSOS

Básica

Bulding Java Programs; Stuart Reges, Marty Stepp; Pearson; 2010

El lenguaje de programación Java; K. Arnold, J. Gosling y D. Holmes; Addison-Wesley; 2001

Java SE 8 for programmers. P. Deitel y H. Deitel; Prentice Hall 2014

Java 2, JDK 7. Ivor Horton's. 2011

Java 2 v5.0; H. Schildt; Anaya; 2005

Programación Orientada a Objetos con Java; Fco. Durán, Fco. Gutiérrez, E. Pimentel; Thomson; 2007

Complementaria

A Programmer's Guide to Java Certification; K. Mughal, R. Rasmunssen; Addison-Wesley; 2004

Construcción de Software Orientado a Objetos; B. Meyer; Prentice- Hall; 1999

Java SE 8 for Programmers; Paul Deitel, Paul Deitel; Prentice Hall; 2015;

The Java tutorial: a short course on the basics; M. Campione, K. Walrath y A. Huml; Addison-Wesley; 2013; Disponible en http://java.sun.com/docs/books/tutorial

The Java tutorial continued: the rest of the JDK; K. Walrath, M. Campione. y A. Huml; Addison-Wesley; 2013

The JFC Swing tutorial: a guide constructing GUIs; K. Walrath, M. Campione. y A. Huml; Addison-Wesley; 2013

DISTRIBUCIÓN DEL TRABAJO DEL ESTUDIANTE

ACTIVIDAD FORMATIVA PRESENCIAL

Descripción	Horas	Grupo grande	Grupos reducidos
Lección magistral	41,4	☑	☐
Prácticas en laboratorio	18,6	☐	☑
TOTAL HORAS ACTIVIDAD FORMATIVA PRESENCIAL	**60**		
TOTAL HORAS ACTIVIDAD FORMATIVA NO PRESENCIAL	**75**		
TOTAL HORAS ACTIVIDAD EVALUACIÓN	**15**		
TOTAL HORAS DE TRABAJO DEL ESTUDIANTE	**150**		