

320030 - CP - Complementos de Programación

Unidad responsable:	205 - ESEIAAT - Escuela Superior de Ingenierías Industrial, Aeroespacial y Audiovisual de Terrassa
Unidad que imparte:	723 - CS - Departamento de Ciencias de la Computación
Curso:	2019
Titulación:	GRADO EN INGENIERÍA DE SISTEMAS AUDIOVISUALES (Plan 2009). (Unidad docente Optativa) GRADO EN INGENIERÍA DE DISEÑO INDUSTRIAL Y DESARROLLO DEL PRODUCTO (Plan 2010). (Unidad docente Optativa) GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA (Plan 2009). (Unidad docente Optativa) GRADO EN INGENIERÍA DE TECNOLOGÍA Y DISEÑO TEXTIL (Plan 2009). (Unidad docente Optativa) GRADO EN INGENIERÍA MECÁNICA (Plan 2009). (Unidad docente Optativa) GRADO EN INGENIERÍA QUÍMICA (Plan 2009). (Unidad docente Optativa) GRADO EN INGENIERÍA ELÉCTRICA (Plan 2009). (Unidad docente Optativa)
Créditos ECTS:	6
Idiomas docencia:	Catalán, Castellano

Profesorado

Responsable:	Jordi Marco
Otros:	Pepa López

Competencias de la titulación a las cuales contribuye la asignatura

Transversales:

1. APRENDIZAJE AUTÓNOMO - Nivel 3: Aplicar los conocimientos alcanzados en la realización de una tarea en función de la pertinencia y la importancia, decidiendo la manera de llevarla a cabo y el tiempo que es necesario dedicarle y seleccionando las fuentes de información más adecuadas.
2. COMUNICACIÓN EFICAZ ORAL Y ESCRITA - Nivel 3: Comunicarse de manera clara y eficiente en presentaciones orales y escritas adaptadas al tipo de público y a los objetivos de la comunicación utilizando las estrategias y los medios adecuados.
3. TRABAJO EN EQUIPO - Nivel 3: Dirigir y dinamizar grupos de trabajo, resolviendo posibles conflictos, valorando el trabajo hecho con las otras personas y evaluando la efectividad del equipo así como la presentación de los resultados generados.

320030 - CP - Complementos de Programación

Metodologías docentes

Tipos de sesiones:

- Sesiones presenciales de exposición de los contenidos y de trabajo práctico en el laboratorio.
- Trabajo autónomo de estudio y realización de ejercicios.
- Preparación y realización de actividades evaluables en grupo.

En las sesiones de exposición de los contenidos el profesor introducirá las bases teóricas de la materia, conceptos, métodos y resultados ilustrándolo con ejemplos convenientes para facilitar su comprensión.

Las sesiones de trabajo práctico en el aula serán de tres clases:

- a) Sesiones en las que los estudiantes seguirán un guión de prácticas con ejercicios intercalados y el profesor resolverá las dudas que vayan surgiendo.
- b) Sesiones de presentación de trabajos realizados en grupo por parte de los estudiantes.
- c) Sesiones de exámenes

Los estudiantes, de forma autónoma deberán estudiar para asimilar los conceptos, resolver los ejercicios propuestos ya sea con lápiz y papel o utilizando el ordenador.

Los estudiantes elaborarán un proyecto de programación en grupos de dos o tres.

Objetivos de aprendizaje de la asignatura

La asignatura de Complementos de Programación tiene como objetivo fundamental preparar al alumno para:

- Utilizar un nuevo paradigma de programación: Orientación a Objetos.
- Utilizar programación visual para el diseño de interfaces gráficas.
- Trabajar con eventos.
- Comprender y utilizar clases de librería.

Para superar la asignatura, el alumno debe ser capaz de:

- Realizar implementaciones en Java para resolver problemas de envergadura media haciendo servir interfaces gráficas, orientación a objeto y librerías de clases.

Horas totales de dedicación del estudiantado

Dedicación total: 150h	Horas grupo grande:	0h	0.00%
	Horas grupo mediano:	0h	0.00%
	Horas grupo pequeño:	60h	40.00%
	Horas actividades dirigidas:	0h	0.00%
	Horas aprendizaje autónomo:	90h	60.00%

320030 - CP - Complementos de Programación

Contenidos

<p>Tema 1. Clases y Objetos</p>	<p>Dedicación: 30h Grupo pequeño/Laboratorio: 12h Aprendizaje autónomo: 18h</p>
<p>Descripción: Especificación e implementación. Miembros: atributos y métodos. Accesibilidad a miembros. Tipo de métodos: constructores, destructores, consultores, modificadores y operadores. Sobrecarga de métodos. Gestión dinámica de la memoria.</p> <p>Actividades vinculadas: Laboratorio: - Desarrollo de programas en relación con el tema.</p> <p>Objetivos específicos: - Determinar los constructores necesarios para una clase. - Distinguir los diferentes tipos de métodos. - Implementar constructores de clases simples y agregadas. - Detectar errores de acceso a miembros. - Distinguir el método escogido en una llamada sobrecargada. - Implementar correctamente un TAD. - Crear objetos en memoria dinámica.</p>	
<p>Tema 2. Herencia y Polimorfismo</p>	<p>Dedicación: 42h Grupo pequeño/Laboratorio: 18h Aprendizaje autónomo: 24h</p>
<p>Descripción: Clases derivadas. Acceso a miembros. Métodos constructores en clases derivadas. Jerarquías de clases. funciones polimórficas. Tablas heterogéneas. Concepto de interfaz.</p> <p>Actividades vinculadas: Laboratorio: - Desarrollo de programas en relación con el tema.</p> <p>Objetivos específicos: - Definir clases derivadas. - Implementar constructores de clases derivadas. - Dado un programa, realizar una modificación usando la herencia.</p>	

320030 - CP - Complementos de Programación

<p>Tema 3. programación visual</p>	<p>Dedicación: 18h Grupo pequeño/Laboratorio: 6h Aprendizaje autónomo: 12h</p>
<p>Descripción: Diseño de interfaces gráficas. Programación con eventos.</p> <p>Actividades vinculadas: Laboratorio: Desarrollo de programas en relación con el tema.</p> <p>Objetivos específicos: - Utilizar diferentes clases visuales. - Trabajar con los eventos más habituales. - Diseño de interfaz gráfica con el modelo vistacontrolador.</p>	
<p>Tema 4. Estructuras de datos</p>	<p>Dedicación: 60h Grupo grande/Teoría: 24h Aprendizaje autónomo: 36h</p>
<p>Descripción: Librerías de estructuras de datos: Java Collections Framework (JCF).</p> <p>Actividades vinculadas: Laboratorio: - Desarrollo de programas en relación con el tema.</p> <p>Objetivos específicos: - Escoger la estructura de datos adecuada para un problema dado. - Insertar, borrar y modificar los elementos de cualquier estructura. - Buscar y recorrer los elementos en cualquier estructura eficientemente. - Utilizar iteradores. - Desarrollar programas utilizando la JCF.</p>	

Sistema de calificación

Examen parcial: 20%
Examen final: 30%
Laboratorio: 20%
Resolución de problemas: 10%
Proyecto: 20%

Bibliografía

Básica:

Horstmann, C. S.; Cornell, G. Java 2. Vol 1, Fundamentos. Madrid: Prentice Hall, 2003. ISBN 8420537001.