

1) Nombre de la asignatura: Desarrollo de Software para Ingeniería Eléctrica

2) Materia: Informática.

3) Créditos: 10.

4) Objetivo de la asignatura.

El objetivo central del curso es introducir métodos y herramientas modernos para el desarrollo de aplicaciones complejas de software (o de sistemas que constan de hardware y software).

La intención del curso es generalizar la enseñanza de la orientación a objetos a nivel de la carrera de ingeniería eléctrica, haciendo hincapié en los conceptos esenciales más que en un lenguaje de programación en particular.

5) Metodología de enseñanza.

Ubicación en la Carrera

La asignatura estará ubicada en el séptimo semestre de la carrera, y tendrá carácter de opcional (aunque sería deseable que al menos los perfiles más próximos a estos temas la incluyan como obligatoria).

Estructura

Se dictarán 4 horas semanales de teórico y dos de ejercicios. Además, habrá algunos talleres donde los estudiantes realizarán pequeñas aplicaciones.

Metodología de Enseñanza

El curso utilizará una metodología de enseñanza basada en el estudio de casos. Se pretende lograr que el estudiante vislumbre en la práctica las ventajas de tomar un enfoque sistemático para el desarrollo de software, a través de experimentar y comparar los resultados de usar (y de no usar) una metodología de desarrollo de software.

El énfasis del curso no será en la enseñanza de un lenguaje de programación en particular, pero una parte importante del curso de ejercicios estará destinado a resolver problemas utilizando un lenguaje de programación, que denominaremos LPOOC en la tabla de contenidos de la asignatura, para resaltar el hecho de que dicho lenguaje seguramente variará con el paso del tiempo.

En el curso práctico, se introducirá el LPOOC y se propondrán pequeños programas a desarrollar en su casa por el estudiante (usando herramientas de dominio público). También existirán horas del laboratorio de software disponible para aquellos estudiantes que no dispongan de computadora en su casa.

Hacia la segunda mitad del semestre, en el práctico se plantearán problemas "reales", y el estudiante deberá planificar y llevar a cabo el desarrollo de soluciones para los mismos.

Limitación del número de alumnos

Existirá un cupo que dependerá de la cantidad de docentes que sean asignados para dictarla. Si el número de inscriptos en estas condiciones no supera el cupo dispuesto, las vacantes podrán ser completadas por estudiantes que ya la hayan cursado y reprobado así como por estudiantes de otras carreras que deseen cursarla.

6) Temario.

Primera Parte: Orientación a objetos

- Introducción: Hardware y software. Categorías de software: de sistema, de desarrollo, de aplicación. Evolución de los lenguajes y conceptos de programación. Ciclo de vida del software: especificación, análisis, diseño, codificación, prueba, producción, mantenimiento. (1 semana)
- Conceptos Básicos: Clases, Objetos/Instancias, Atributos, Operaciones, Constructor, Visibilidad, Eventos. Ejemplos en LPOOC (2 semanas)
- Composición y Herencia, Polimorfismo, Generalización y Especialización, Interfaces. Ejemplos en LPOOC (1.5 semanas)
- Aplicación de los conceptos en casos de estudio. (1 semana)

Segunda Parte: Ingeniería de Software

- Proceso de Desarrollo de Software. Etapas y tareas. (0.5 semana)
- Etapas iniciales del desarrollo de software: Ingeniería de requerimientos orientada a objetos. Técnicas de análisis: casos de uso y modelado orientado a objeto utilizando UML. Especificación de requerimiento de acuerdo al estándar IEEE/ANSI 830. (2 semanas)
- Caso de Estudio (0.5 semanas)
- Diseño orientado a objeto. Patrones de Diseño (1.5 semanas)
- Testeo de aplicaciones orientadas a objeto (0.5 semanas)

7) Bibliografía.

- Bruce Eckel: Thinking in Java. 2nd Edition. ISBN: 0130273635.
- Erich Gamma, Richard Helm, RAIfh, Johnson, John Vlissides: Design Patterns: element of reusable object-oriented software. ISBN: 0-201-63361-2.
- Craig Larman: Applying UML and patterns: an introduction to object-oriented analysis an design. ISBN: 0-13-748880-7.
- Campione, M., K. Walrath: The Java Tutorial, 2nd Edition. Object-Oriented Programming for the Internet (97). ISBN: 0201310074.
- Davis, A.: Software Requirements. Objects, Functions and States, Prentice Hall (93). ISBN: 013805763X.
- Flanagan, D.: Java in a Nutshell : A Desktop Quick Reference, 2nd Edition, O'Reilly & Associates (98). ISBN: 156592262X.
- Jacobson, I. et al.: Object-Oriented Software Engineering: A Use Case Driven Approach (Addison-Wesley Object Technology Series) (92). ISBN: 0201544350.
- Rumbaugh, J.:et al.: Object-Oriented Modeling and Design, Prentice Hall (91). ISBN: 0136298419

8) Conocimientos previos exigidos y recomendados.

Previatura

Para aprovechar el curso, los alumnos deberán poseer conocimientos básicos de informática, por lo que se sugiere que la asignatura "Computación" (o equivalente) sea considerada como previa a esta asignatura.

ANEXO

Modalidad del curso y procedimientos de evaluación

La aprobación de la asignatura se realizará mediante el régimen de parciales exclusivamente, no existiendo una posterior instancia de examen. Se realizarán tres parciales, el primero y el tercero durante las semanas provistas para ello. El segundo parcial será realizado por cada estudiante con una computadora, donde el estudiante deba escribir, compilar y ejecutar un pequeño programa. Para aprobar el curso se deberá obtener más del 60% de la suma ponderada de los tres parciales.

Se prevé además un parcial complementario para aquellos estudiantes que obtengan más de 50% y menos de 60% del total de puntos, con el cual podrán aprobar la asignatura con el puntaje mínimo.

Quien no apruebe, deberá recurrar, sujeto a la existencia de cupo.

POR RESOLUCION DEL CONSEJO DE FACULTAD DE FECHA

30.04.2001 SEGUN EXP. 060180-000320-01