

## **Programa**

1) **Nombre de la asignatura.** Programación orientada a objetos

2) **Créditos.** 7

3) **Objetivo de la asignatura.**

El objetivo del curso es generalizar la enseñanza de la programación orientada a objetos (POO) a nivel de la carrera de ingeniería eléctrica, haciendo hincapié en los conceptos esenciales más que en un lenguaje de programación en particular.

4) **Metodología de enseñanza.**

### **Estructura**

Se dictarán 4 horas semanales de teórico y 2 de ejercicios. Durante el curso, los estudiantes realizarán además un proyecto mediano que será seguido mediante reuniones de supervisión con los docentes.

### **Metodología de Enseñanza**

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

En el curso práctico, se propondrán pequeños programas a desarrollar por los estudiantes.

Se planteará un proyecto obligatorio de mediano porte que los estudiantes resolverán en forma incremental a lo largo del curso (usando herramientas de dominio público). Este trabajo será supervisado por los docentes y se evaluará en entregas parciales que deberán cumplir los estudiantes.

5) **Temario.**

- Programación y Lenguajes (1 semana)  
Evolución de los lenguajes y conceptos de programación. Ciclo de vida del software: especificación, análisis, diseño, codificación, prueba, producción, mantenimiento
- Introducción a POO (1 semana)  
Paradigmas de programación, paradigma POO, Clases, objetos
- Características del LPOO: (1 semana)  
Sentencias, Control de flujo
- Conceptos Básicos de POO: (2 semanas)  
Clases, Objetos/Instancias, Atributos, Operaciones, Constructor, Visibilidad, Eventos. Ejemplos en LPOO (2 semanas)
- Composición y Herencia (2 semanas)  
Polimorfismo, Generalización y Especialización, Interfaces. Ejemplos en LPOO
- Aplicación de los conceptos en casos de estudio. (1 semana)

## ANEXO

### Materia.

Informática.

### Previatura

Se sugiere que para cursar esta asignatura, se requiera la aprobación de los cursos de las siguientes asignaturas: Cálculo 1, Geometría y Álgebra Lineal 1, Física 1, Programación 1.

#### 1) Cronograma tentativo

En el temario del curso se indican aproximadamente los tiempos que se dedicarán a cada una de las partes que componen el curso.

#### 2) Modalidad del curso y procedimientos de evaluación

La asignatura se aprobará mediante el método de parciales y la entrega de un obligatorio exclusivamente, no existiendo una posterior instancia de exámen. Se realizarán dos parciales y dos (o tres) entregas del trabajo obligatorio. Para aprobar el curso se deberá obtener más del 60% de la suma ponderada de todas las pruebas (parciales y entregas obligatorias). Además se deberá superar un mínimo del 25 % de cada prueba. Los parciales evaluarán conceptos teóricos de POO, específicos sobre el LPOO y vinculados al trabajo obligatorio.

Quien no apruebe deberá recurrir completamente la asignatura, sujeto a la existencia de cupo.

- Casos de reprobación
  - No se supera el mínimo en alguna de las pruebas.
  - No se supera el 60% de la suma ponderada de todas las pruebas.

#### Estimación de horas y créditos

17 clases de teórico: 2 hrs = 34 hrs

8 clases de práctico: 2 hrs = 16 hrs

parciales: 3 hrs

total horas presenciales: 52 hrs

Horas estimadas de estudio

horas estudio complementario: 16 hrs

horas de trabajo obligatorio: 40 hrs

Total: 109 hrs, créditos: 7

## 6) **Bibliografía.**

### Bibliografía Tentativa

- Básica
  - Thinking in Java, Bruce Eckel
  - Thinking in C++, Bruce Eckel
  
- Complementaria
  - Campione, M., K. Walrath: The Java Tutorial, 2<sup>nd</sup> Edition. Object-Oriented Programming for the Internet (97)
  - Davis, A.: Software Requirements. Objects, Functions and States, Prentice Hall (93)
  - Flanagan, D.: Java in a Nutshell : A Desktop Quick Reference, 2<sup>nd</sup> Edition, O'Reilly & Associates (98)
  - Humphrey, W.: A Discipline for Software Engineering (SEI Series in Software Engineering) (95)
  - Humphrey, W.: Managing the Software Process (89)
  - Jacobson, I. et al.: Object-Oriented Software Engineering : A Use Case Driven Approach (Addison-Wesley Object Technology Series) (92)
  - Pressman, R: Software Engineering : A Practitioner's Approach, 4th edition (1996)
  - Rumbaugh, J. et al.: Object-Oriented Modeling and Design, Prentice Hall (91)
  - Thayer, R., Yourdon, E.: Software Engineering Project Management 2nd edition (October 1997) IEEE Computer Society; ISBN: 0818680008
  - The Java Language Specification, (3rd Edition) James Gosling, Bill Joy, Guy Steele, Gilad Bracha, ISBN-13: 978-0321246783
  - Como programar en C++, H.M.Deitel y P.J. Deitel. Prentice Hall

## 7) **Conocimientos previos exigidos y recomendados.**

Conocimientos básicos de programación estructurada.

## Calendario tentativo

clase	teórico	práctico	obligatorio	
1	U1: Presentación. Programación y Lenguajes	No hay práctico		
2	U1: Programación y Lenguajes. U2: intro OOP			
3	U2: Introducción a OOP. Objeto y Clase. U3: Características de LPOO. HolaMundo.			
4	U3: Sentencias. Control de flujo	Hoja1: Sintaxis		
5	U4: Control de flujo. Objetos.	Hoja2: Objetos básico		
6	U4: Objetos.			
7	U4: Objetos.	Hoja2: Clases avanzado		
8	U5: Clases.			
9	U5: Clases.	Hoja3: Colecciones		
10	U6: Paquetes. U6.5: Colecciones			
11	U7: Herencia.	Hoja4: Inic., package, classpath		
12	U7: Herencia.			
13	U8: Interfaces.	Hoja5: Herencia (I)		
14	U9: Excepciones.			
	SEMANA DE TURISMO			
15	U9: Excepciones.	Hoja5: Herencia (II)		
16	U10: Entrada/Salida		Primera entrega	
	<b>PRIMEROS PARCIALES</b>			
17	Ejemplos – casos de estudio	Hoja6: Excepciones		
			Segunda entrega	
	<b>SEGUNDOS PARCIALES</b>			

APROB. RES. CONSEJO DE FAC. ING.

de fecha 16.2.12 Exp. 060180-003177-11