UNIVERSIDAD DIEGO PORTALES
Facultad de Ingeniería
Escuela de Ingeniería Informática

PROGRAMA DE ESTUDIOS 2004

ASIGNATURA	:	**COMPUTACIÓN I**
Código	:	INF 2100
Pre-requisito	:	Admisión
Requisito de	:	Computación II
N ° sesiones semanales	:	2 de Cátedra
	:	1 de Ayudantía o Laboratorio

I OBJETIVOS GENERALES

Adquirir conceptos y elementos que permitan resolver problemas con la ayuda del computador, usando un paradigma orientado a objetos.

II OBJETIVOS ESPECÍFICOS

Al finalizar el curso el alumno deberá ser capaz de:

- Conocer los conceptos básicos del computador, lenguajes de programación, programas traductores.
- Analizar problemas para determinar objetos, clasificarlos en clases, describir los métodos que se deben usar para manipularlos, y caracterizar como representar su estado.
- Aprender los fundamentos y conceptos básicos de la Programación Orientada al Objeto: clases, objetos, abstracción, herencia, encapsulamiento y polimorfismo.
- Programar algoritmos para implementar la solución de problemas por medio de un lenguaje de programación orientado al objeto.
- Aprender a diseñar nuevas clases y utilizar clases predefinidas, leer código y adaptarlo de acuerdo a requerimientos específicos.
- Aprender un lenguaje orientado a objetos.

III CONTENIDOS

1. FUNDAMENTOS DE OBJETOS Y CLASES

Objetos y Clases. Creación de objetos. Invocación de métodos. parámetros y tipos de datos. Múltiples instancias de una clase. El estado de un objeto. Interacción entre objetos. Código fuente. Estudio de un caso. Valores de retorno. Objetos como parámetros.

2. LA DEFINICIÓN DE CLASES

Presentación de un caso de estudio. La definición de clases. Campos, constructores y métodos. Paso de parámetros. Asignación. Métodos de acceso. Métodos mutantes. Como imprimir desde un método. Revisión del caso de estudio. Toma de decisiones. Variables locales. Campos, parámetros y variables locales.

3. INTERACCION ENTRE OBJETOS

Presentación de un caso de estudio. Abstracción y modularización. Implementación del caso de estudio. Diagramas de clases vs. Diagramas de objetos. Tipos primitivos y tipos de objetos. Objetos que crean objetos. Múltiples constructores. Invocación de métodos. Uso de un debugger para probar el programa.

.

4. AGRUPACION DE OBJETOS

Agrupación de objetos en colecciones. Presentación de un caso de estudio. Introducción a bibliotecas de clases. Como se estructuran los objetos en las colecciones. Numeración en colecciones. Proceso de una colección completa, while, iteradores, índices. Colecciones de tamaño fijo.

5. USO DE ELEMENTOS AVANZADOS

Documentación de bibliotecas. Uso de JavaDoc. Uso de números aleatorios (Random). Uso de mapas. Uso de conjuntos. Manejo de strings. Paquetes e importación. Encapsulamiento (Public vs. Private). Variables de clase y constantes.

6. OBJETOS DE BUEN COMPORTAMIENTO

Pruebas y debugging. Pruebas unitarias. Automatización de las pruebas. Modularización e interfaces. Debugging. Comentarios y estilo de codificación. Revisiones manuales del código. Estrategias de pruebas.

7. DISEÑO DE CLASES

Acoplamiento y cohesión. Código duplicado. Extensiones. Uso del encapsulamiento para reducir el acoplamiento. Diseño por responsabilidades. Cohesión. Refactoring. Guías de diseño.

IV METODOLOGÍA

Se contempla la realización de 1 sesión semanal en Laboratorio y una sesión de cátedra en aula de clases, además se dispone de tutoría en Laboratorio, donde se aplicará la técnica de estudio de casos, en las que las materias del curso se irán desarrollando en base a los ejemplos aplicados que en cada sesión se planteen.

El curso será apoyado con apuntes de clases almacenados en un servidor para acceso vía Internet, en el que se describirán las materias y los ejercicios que deberán realizarse clase a clase.

Evaluación de la teoría

Se contempla la realización de dos solemnes, 4 controles, 2 tareas y un examen.

Evaluación de los laboratorios

Para aprobar la asignatura el alumno DEBE haber aprobado el laboratorio, donde la asistencia al 100% de las experiencias es una condición necesaria, pero no suficiente.

Evaluación de la asignatura

- La nota de presentación a examen (NP) estará compuesta de 60% nota de Solemne más 40% promedio de tareas/laboratorios.
- La nota final de la asignatura (NF) tendrá una ponderación de 70% nota final de cátedra y 30% de examen.
- Para aprobar el curso debe tenerse que NF $\geq$ 4.0 y para presentarse a Examen NP $\geq$ 3.5

V BIBLIOGRAFÍA

- Barnes, David and Kolling, Michael, *Objects First with Java*, 2nd edition, Pearson Education, 2005.

Bibliografía Complementaria.

- Eckel, Bruce: *Thinking in Java*, 3ª Edición. (Documento HTML disponible en Internet, en http://euler.udp.cl/TIJ/)
- Varios autores de la Universidad de Navarra: *Aprenda Java como si estuviera en primero*. (PDF disponible en Internet http://www.abcdatos.com/tutoriales/tutorial/l7041.html)

UNIVERSIDAD DIEGO PORTALES
ESCUELA DE INGENIERÍA CIVIL

Programa de Asignatura cursado por: ***MATIAS ANTONIO ARMAZA GODOY, RUT.16.661.299-3*** durante el Segundo semestre del 2006, obteniendo una calificación de 5,5 (CINCO COMA CINCO)

XIMENA GEOFFROY W.
SECRETARIA DE ESTUDIOS
ESCUELA INGENIERIA INFORMATICA

PAUTAS ETICAS BASICAS

El plagio es el uso de las ideas o trabajo de otra persona sin el adecuado consentimiento. El plagio puede ser intencional o no. El plagio intencional es el claro intento de hacer pasar el trabajo o ideas ajenas como el suyo propio para su beneficio. El plagio no intencional puede ocurrir si Ud. no conoce el mecanismo adecuado de referenciar la fuente de sus ideas e información. Si no está seguro de los métodos aceptados para referenciar, debería consultar con su profesor, tutor o personal de biblioteca.

El plagio comprobado es una actitud que puede resultar en severas sanciones disciplinarias y/o en la exclusión de la Universidad (Articulo 44, Reglamento del Estudiante de Pregrado).