UNIVERSIDAD
ANDRES BELLO

I. IDENTIFICACIÓN

CARRERA	Ingeniería en Gestión Informática
ASIGNATURA	Programación
CÓDIGO ASIGNATURA	INF1201
REQUISITOS	INGRESO
CO-REQUISITOS	SIN CORREQUISITO
RÉGIMEN	Diurno y vespertino
CARÁCTER	Teórico y laboratorio
NIVEL	1er semestre
DURACIÓN	Semestral
CRÉDITOS	6

II. DESCRIPCIÓN

El estudiante aprenderá a analizar un problema y plantear una solución racionalmente, utilizando programación orientada a objetos, aplicando los fundamentos de objetos y clases mediante un ambiente IDE que permita el desarrollo y depuración de código del lenguaje. Además, será capaz de describir y entender el proceso de compilación y ejecución.

III. OBJETIVOS

Objetivo General

- Desarrollar el pensamiento algorítmico para resolver problemas de ingeniería mediante el uso de un lenguaje de programación estructurado.

Objetivos Específicos

- Analizar problemas identificando sus componentes relevantes.
- Resolver problemas mediante la descomposición de estos en otros más pequeños.
- Conocer la lógica y metodología para el desarrollo de aplicaciones utilizando programación estructurada.
- Resolver problemas mediante el desarrollo de algoritmos.
- Traducción de algoritmos a programas computacionales.
- Aplicar las instrucciones básicas de un lenguaje de programación estructurado: secuencia, sentencias de selección y sentencias de repetición.
- Diseño de programas modulares.

IV. CONTENIDOS

Unidad 1: Introducción

En esta unidad se presentará los problemas y las formas de abordarlos, conocer que es un algoritmo y las partes que lo forman. Se introducirá al alumno al pseudo código y a los diferentes modelos de programación. Adicionalmente se explicarán los diferentes tipos de licencias de software que existen.

Unidad 2: **Programación estructurada**

Estudiar conceptos relacionados con la técnica de programación estructurada utilizando las estructuras de secuencias, selección e iteración.

Unidad 3: Funciones y módulos

Durante esta unidad el alumno estudiará como diseñar y contruir programas modulares que permitan encapsular el código y su fácil reutilización.

Unidad 4: **Programación orientada a objetos**

La unidad explica el paradigma de programación orientada a objetos, donde el alumno comprenderá como transformar objetos del mundo real a objetos en sus programas computacionales. Se verán conceptos como herencia, abstracción, encapsulamiento y polimorfismo.

Unidad 5: **Manipulación de archivos**

Lectura y escritura de archivos de texto planos.

Unidad 6: **Estructuras de datos simples**

Se hará estudio de estructuras de datos básicas (como arreglos) utilizando el paradigma de programación orientada a objetos.

Unidad 7: **Procesamiento de texto**

Procesamiento de texto, reemplazo de caracteres, separadores, búsqueda de patrones. Concepto de expresiones regulares.

V. EVALUACIÓN

La nota de presentación a examen se calculará como sigue:

Control 1 = 30%
Control 2 = 30%
Laboratorio = 40%

Si la nota de presentación a examen es igual o superior a 5.0, el alumno o alumna podrá eximirse de la rendición del examen de la asignatura, siempre y cuando no presente notas parciales bajo 4,0.

Los alumnos que no se eximan calcularán su calificación final según lo siguiente:

Nota de presentación * 0.7 + Nota de examen * 0.3

Las fechas de evaluaciones serán informadas oportunamente.

La asistencia requerida para aprobar el curso es 75%, de forma contraria se reprobará el curso con nota final igual a:

MÍN(nota final, 3.9)

Finalmente la aprobación del curso es por separado para controles y laboratorio. O sea, se requiere nota igual o superior a 4.0 tanto en el promedio de controles como en el promedio de laboratorio para poder aprobar el curso. En caso contrario la nota de presentación y por ende la nota de examen será:

MÍN(promedio controles, promedio laboratorio)

VI. BIBLIOGRAFÍA

Bibliografía Obligatoria

- Kernighan, B., Ritchie, D. (1988). *The C Programming Language, 2nd Edition*.
- Ceder, V. (2010). *The Quick Python Book*. Londres: Manning.
- Downey, A. (2009). *Python for Software Design*. Londres: Cambridge.
- Campbell, J., Gries, P., Montojo, J., Wilson, G. (2009). *Practical Programming: An Introduction to Computer Science Using Python*. Estados Unidos: The Pragmatic Bookshelf.
- Lutz, M. (2009). *Learning Python*. Estados Unidos: O'Reilly Media.

Bibliografía Extra

- Joyanes, L. (2003). *Fundamentos de programación: Algoritmos, estructura de datos y objetos*. Madrid: McGraw-Hill.
- Joyanes L. Fundamentos de Programación - Libro de Problemas. Mc GrawHill.
- Lemay, L. Perkins, C. (1996). Aprendiendo Java en 21 días. Prentice Hall.
- Deitel, H., Deitel, P. Como programar en C++. Pearson educación.
- Deitel, H., Deitel, P. Como programar en C#. Pearson educación.

Recursos en línea

- Curso de Programación en USM: http://progra.usm.cl
- Documentación oficial Python: http://docs.python.org
- Python no muerde: http://nomuerde.netmanagers.com.ar
- Compilador online: http://profesores.ing.unab.cl/~delaf/ide