



Programa de la Asignatura:

Lenguajes de Programación I



Código: 745

Carrera: **Ingeniería en Computación** Plan: **2008** Carácter: **Obligatoria**
Unidad Académica: **Secretaría Académica** Curso: **Primer Año – Segundo cuatrimestre**
Departamento: **Ingeniería** Carga horaria total: **60 hs.** Carga horaria semanal: **4 hs.**
Formación Experimental: **30 %** Formación teórica: **40 %** Formación práctica: **30 %**

Materias Correlativas Obligatorias

- **Estructuras de Datos I (cód. 743)**
- -----
- -----

Cuerpo Docente

Fontdevila, Diego
Videla, Lucas

Índice

- Fundamentación pág. 2
- Encuadre y articulación de la asignatura pág. 2
 - Encuadre dentro del Plan de Estudios pág. 2
 - Articulación Horizontal pág. 3
 - Articulación Vertical pág. 3
- Objetivos pág. 3
 - Objetivo General pág. 3
 - Objetivos Específicos pág. 3
- Contenidos mínimos pág. 3
- Programa analítico pág. 3
- Bibliografía básica pág. 5
- Bibliografía de consulta pág. 5
- Metodología del aprendizaje pág. 5
 - Desarrollo de la asignatura pág. 5
 - Dinámica del dictado de las clases pág. 6
 - Trabajos prácticos pág. 7
- Metodología de evaluación pág. 7
- Planificación pág. 8
- Información de versiones pág. 8

AÑO ACADÉMICO 2013

ÚLTIMA REVISIÓN 07/06/2013

Firma Docente

Firma Coordinador

1. FUNDAMENTACION

La asignatura está incluida entre las materias que brindará a los alumnos los conocimientos necesarios para poder desarrollar soluciones programáticas, utilizando un lenguaje de programación (en este caso, Java) para delinear la forma de la resolución de los problemas planteados.

Con la intención de continuar con el aprendizaje tomado durante el primer cuatrimestre con Estructuras de Datos I, y debido a la necesidad de profundizar los contenidos teóricos y fundamentalmente prácticos de la programación, esta asignatura se propone aumentar la carga práctica y la relación del alumno con el lenguaje de programación.

Asimismo se plantean problemas más cercanos a los entornos laborales, además de elevar el nivel de dificultad de la práctica. Conforme transcurre el cuatrimestre se comienzan a dictar contenidos de más bajo nivel que los aprendidos durante la materia anterior.

Este hecho abre el campo a una infinidad de posibilidades al momento de conjugar las piezas y de comprender el funcionamiento de la programación como herramienta para la resolución de problemas del manejo de la información.

Siguiendo con el enfoque de Estructuras de Datos I, se aplica el Currículo Invertido de Meyer, aunque en este caso se comienza a desglosar la estructura general de la programación para pasar a tomar foco en los componentes de menor nivel, y la sintaxis y semántica del lenguaje.

2. ENCUADRE Y ARTICULACIÓN DE LA ASIGNATURA

Encuadre dentro del Plan de Estudios

La asignatura está ubicada en la currícula de la carrera en la parte inicial de la misma que corresponde al "Ciclo Profesional".

En esa etapa, los alumnos continúan con la adquisición de experiencia en el área de la programación.

Requieren los conceptos de lógica, estructuras de control de flujo, nociones de programación orientada a objetos, modularidad y contrato.

Esta materia es correlativa de Estructuras de Datos I.

Articulación Horizontal

En cuanto a la articulación horizontal se ha planeado que esta asignatura se curse en simultáneo con Álgebra II, Matemática Discreta II y Física II, de modo que el alumno pueda integrar una perspectiva profunda y abarcadora de técnicas generales de resolución de problemas complejos, aprovechando conocimientos nóveles que poseen aplicación directa en consignas clásicas.

Articulación Vertical

Esta asignatura se articula verticalmente con Estructura de Datos I que brinda los conocimientos básicos del funcionamiento de un lenguaje de programación, y los lineamientos de la algoritmia como base para continuar el aprendizaje de nuevas técnicas.

Por otra parte, a esta materia le seguirá en un curso superior la asignatura Lenguaje de Programación II que proveerá al alumno de los conocimientos más avanzados y complejos de la programación orientada a objetos.

3. OBJETIVOS

Objetivo General

La cátedra se ha fijado como “objetivo cognoscitivo” de esta materia, **“Lograr que los alumnos profundicen sobre los principios de la resolución de problemas, y las técnicas más comunes para ello; adicionalmente se espera que amplíen sus conocimientos sobre la programación orientada a objetos, y que comiencen a refinar su forma de trabajo acorde al estado del arte de la profesión”.**

Objetivos Específicos

Luego de cursar esta asignatura el alumno deberá dominar los siguientes temas:

- Desarrollar soluciones avanzadas con objetos
- Conocer e implementar algoritmos de ordenamiento, búsqueda, mezcla y corte de control
- Comprender la necesidad y uso de herramientas para probar el software desarrollado
- Utilizar herramientas provistas por el lenguaje y bibliotecas de componentes
- Aplicar técnicas de TDD (Test-Driven Development, Desarrollo guiado por las pruebas)

4. CONTENIDOS MÍNIMOS

Procesadores, sistemas operativos y lenguajes. Abstracción y técnicas de solución de problemas. Construcción de algoritmos. Construcción y depuración de programas. Direccionamientos en memoria con punteros. Programas y módulos construidos usando arreglos. Definición y uso de referencias. Algoritmos mayormente conocidos: ordenamiento y búsqueda.

5. PROGRAMA ANALÍTICO

Unidad 1: Repaso

Modelos. Lógica. Objetos. Algoritmos. Estructuras de control. Resolución de problemas

Unidad 2: Programación por contrato

Concepto. Modelo Eiffel. Modelo Java. Rupturas de contrato.

Unidad 3: Pruebas

Calidad. Pruebas: Concepto. Roles para la confección y ejecución de pruebas. Error. Defecto. Falla. Detalle de pruebas. Tipos de pruebas. Conveniencias. JUnit: Historia, usos. Ejemplo.

Unidad 4: TDD

Concepto. Espíritu. Contraposición al paradigma actual. Ejemplo.

Unidad 5: Arreglos

Datos atómicos. Datos estructurados: Calculables, Asociativos. Masividad del término. Arreglos. Definición. Acceso y recorrido.

Unidad 6: Arreglos multidimensionales

Arreglos. Generalización. Regulares. Irregulares. Inicialización. Recorrido.

Unidad 7: Memoria

Tareas. Modelo de Von Neumann. Componentes. Datos. Información. Datos y Programas. Tipos de Memoria. Volátil. No Volátil. Central. Secundaria. Segmentos de Memoria: CS, DS, SS, ES. Memoria dinámica. Asignación. El operador new. Liberación. Garbage Collection. Esquema de Memoria.

Unidad 8: Algoritmos de Ordenamiento

Concepto de Orden. Clave. Características. Algoritmos. Clasificación: Internos, Externos. Simples. Avanzados. Selección. Burbujeo. Inserción. Mejoras.

Unidad 9: Algoritmos de Búsqueda

Concepto. Secuencial. Binaria. Descubrimiento de mejoras.

6. BIBLIOGRAFÍA BÁSICA

- MEYER , BERTRAND, Construcción de Software Orientada a Objetos, Prentice-Hall, 1985, 2da. Edición 1997.
- MEYER, BERTRAND, Touch of class, Springer, 2009.

7. BIBLIOGRAFÍA DE CONSULTA

- ARNOLD, GOSLING, HOLMES, Holmes, El lenguaje de programación Java, Addison-Wesley, 3ra. Edición 2002.
- ECKEL, BRUCE, Pensando en Java, Prentice-Hall, 2006
- KNUTH, DONALD, The Art of Computer Programming, volumen 3, Sorting and Searching, 1973
- AHO, ALFRED, Estructura de datos y algoritmos, Addison-Wesley, 1988

8. METODOLOGÍA DEL APRENDIZAJE

8.a DESARROLLO DE LA ASIGNATURA

Continuando con la temática propuesta, se aumentará la aproximación a la resolución de problemas, especialmente a aquellos de tipo algorítmico y enfocados en los que involucran algoritmos mayormente conocidos.

Se continuará el uso del “juego didáctico” basado en naves espaciales y comandos básicos. Se agregará el componente de la construcción de piezas propias de código que los alumnos podrán ejecutar y comprobar su correcta ejecución y funcionamiento.

8.b DINÁMICA DEL DICTADO DE LAS CLASES

Para poder llevar adelante los objetivos de la cátedra, se hace un gran énfasis hacia la carga horaria de las aproximaciones a problemas, dando más peso a éstas que a la teoría.

El Profesor a cargo del curso se ocupará en forma personal y semanal del dictado de aquellos temas con un fuerte contenido teórico y que significan conceptos intermedios y poco volátiles en la especialidad. Procederá a describir técnicas, características y propondrá ejemplos.

Asimismo habrá una parte de la clase dedicada a la aplicación de los conceptos vistos durante la parte teórica. Se desarrollarán problemas con creciente nivel de dificultad, y acorde a los temas que se desarrollen.

8.c TRABAJOS PRÁCTICOS

Habrán dos trabajos prácticos durante el cuatrimestre, los cuales tendrán como objetivo resolver un problema algorítmico novedoso, para el cual los alumnos deberán profundizar sobre los conceptos vistos.

Los mismos serán de consigna variante según el año en curso, lo cual ayuda a darle flexibilidad a la cátedra.

9. METODOLOGÍA DE EVALUACIÓN

9.a NORMAS DE EVALUACIÓN.

- El criterio es que la evaluación del alumno es permanente.
- Se tomarán dos exámenes parciales teórico/prácticos pudiendo acceder a un recuperatorio.
- Las notas de los parciales representan los resultados de la evaluación teórico/práctica.
- Los exámenes parciales y sus recuperatorios serán escritos.

9.b RÉGIMEN DE APROBACIÓN DE LA MATERIA.

- Para la aprobación de la materia los alumnos deberán tener los dos parciales aprobados, teniendo la posibilidad de recuperar cada uno de ellos en una única oportunidad adicional, en la fecha acordada con los docentes.
- Además los alumnos deberán aprobar los trabajos prácticos, como condición para la aprobación de la materia.
- Los alumnos que obtengan una nota inferior a cuatro puntos se les asignará la nota insuficiente y deberán recursar la materia.
- Los alumnos que obtengan una nota superior a siete puntos, se les tomará la materia como promocionada, eximiéndolos del examen final.

10. PLANIFICACIÓN

CALENDARIO DE CLASES Y EVALUACIONES	
Semana 1	Unidad 1
Semana 2	Unidad 2
Semana 3	Unidad 3
Semana 4	Unidad 4
Semana 5	Unidad 4 – Práctica
Semana 6	Unidad 5
Semana 7	Unidad 6
Semana 8	Primer Parcial
Semana 9	Unidad 7
Semana 10	Unidad 7 – Parte 2
Semana 11	Trabajo Práctico 2
Semana 12	Unidad 8
Semana 13	Segundo Parcial
Semana 14	Repaso general
Semana 15	Recuperatorios
Semana 16	Cierre de notas
Del al de	FINAL

Información de Versiones	
Nombre del Documento:	Ficha Académica de la asignatura Lenguajes de Programación I
Nombre del Archivo	Lenguajes de Programación I.docx
Documento origen:	
Elaborado por:	Diego Fontdevila
Revisado por:	Lucas Videla
Aprobado por:	Diego Fontdevila
Fecha de Elaboración:	01-02-2013
Fecha de Revisión:	07-06-2013
Fecha de aprobación:	10-06-2013
Versión:	1.0