



Ministerio de Cultura y Educación
 Universidad Nacional de San Luis
 Facultad de Ciencias Físico Matemáticas y Naturales
 Departamento: Informatica
 Area: Area IV: Pr. y Met. de Des. del Soft.

(Programa del año 2012)
 (Programa en trámite de aprobación)
 (Presentado el 03/10/2012 13:07:38)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
PROGRAMACION II	LIC.CS.COMP.	006/05	2012	1° cuatrimestre
PROGRAMACION II	PROF.CS.COMPUT.	06/09	2012	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
NECCO, CLAUDIA MONICA	Prof. Responsable	P.Adj Exc	40 Hs
ALBORNOZ, MARIA CLAUDIA	Auxiliar de Práctico	A.1ra Semi	20 Hs
BERNARDIS, EDGARDO	Auxiliar de Práctico	A.2da Simp	10 Hs
PEREZ, NORMA BEATRIZ	Auxiliar de Práctico	A.1ra Semi	20 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
Hs	3 Hs	3 Hs	3 Hs	9 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
24/03/2012	22/06/2012	15	135

IV - Fundamentación

En este curso se inicia a los alumnos en dos paradigmas: prog. Orientada a Objetos y prog. Declarativa Funcional y se refuerzan y completan los conocimientos y aptitudes que adquirieron en Programación I en el paradigma de la programación estructurada. El contenido mencionado prepara al alumno para poder comparar las características de los distintos modelos formales subyacentes en el desarrollo de programas en los tres paradigmas. Las aptitudes desarrolladas a través de la práctica en esta material, servirán de base tanto para el desarrollo de aplicaciones reales en los distintos paradigmas como para el análisis comparativo formal lenguajes.

También se explora el tema de verificación de programas a través del uso de técnicas de verificación inductivas usando razonamiento lógico para probar ciertas propiedades de algoritmos simples, pero sentando las bases para su posterior tratamiento formal tanto en el ámbito de la ingeniería del software como en el de teoría de la computación.

V - Objetivos / Resultados de Aprendizaje

- Completar y reforzar los conceptos básicos de la programación imperativa.
- Introducir conceptos de la programación avanzada.
- Ejercitar desarrollos de programas basados en métodos estructurados.
- Iniciar el estudio del paradigma de la programación funcional.
- Ejercitar desarrollos de programas basados en el paradigma de la prog. funcional.

- Iniciar el estudio y aplicación del paradigma de la prog. orientada a objetos.
- Ejercitar desarrollos de programas basados en el paradigma de la prog. orientada a objetos.

VI - Contenidos

1. Programación Imperativa:

Encapsulamiento de datos. Tipos de datos abstractos. Uso de punteros.

Implementación de estructuras dinámicas. Recursividad. Introducción a la administración de memoria (principal y secundaria). Entrada/Salida desde memoria secundaria.

- Lenguaje asociado: C.

Características generales. Tipos y operaciones. Compatibilidad vs conversiones de tipos. Arreglos. Estructuras. Uniones. Punteros. Relación entre punteros y arreglos. Aritmética de punteros. Strings. Entrada/Salida en memoria secundaria con formato y sin formato (manejo de Archivos). Administración dinámica de memoria. Clases de almacenamiento. Alcance.

2. Programación Funcional:

Características generales de los lenguajes funcionales. Estilo de programación pointfree vs pointwise. Evaluación perezosa vs Evaluación impaciente.

Transparencia referencial. Concepto de función, definición y aplicación. Pattern matching. Funciones recursivas. Funciones polimórficas. Funciones de orden superior, definición y aplicación. Currificación y aplicación parcial de funciones. Composición de funciones. Manejo de listas.

- Lenguaje asociado: Haskell.

Entorno de trabajo, definición de módulos, uso del intérprete.

Operadores infijos y prefijos. Reglas de precedencia. Prelude de Haskell. Funciones del prelude para manejo de listas y tuplas. Funciones del prelude de orden superior. Definición de funciones recursivas sobre tipos simples y/o estructurados. Manipulación de listas polimórficas. Plegado de listas (foldl, foldr)

3. Programación Orientada a Objetos

Descripción del paradigma de orientación a Objetos. Clases. Objetos. Mensajes.

Conceptos: super, self. Encapsulamiento. Control de accesos. Herencia simple. Herencia múltiple. Sobrecarga y Polimorfismo.

- Lenguaje asociado: Java

Entornos de desarrollo (IDEs) para diferentes plataformas. Descripción gral de Eclipse y NetBeans. Características generales del lenguaje. Tipos de datos básicos. Estructuras de control. Concepto y definición de clase en Java. Objetos del mundo real y abstractos. Concepto de método Java. Desarrollo de código Java básico. Clases con campos, constructores y métodos.

Declaración, creación y destrucción de Objetos. La palabra clave this. Control de acceso. Herencia.

VII - Plan de Trabajos Prácticos

Prácticos de aula:

1. Desarrollo de ejercicios en lenguaje C
2. Desarrollo de ejercicios en Lenguaje Java
3. Desarrollo de ejercicios en Lenguaje Haskell

Prácticos de máquina:

implementaciones concretas de problemas tipo.

Desarrollo y entrega de prácticos de los paradigmas vistos, entre 1 y tres en total según criterio de la cátedra.

VIII - Regimen de Aprobación

- Para regularizar la asignatura:

el alumno debe aprobar dos exámenes parciales o sus correspondientes recuperaciones, y presentar y aprobar en forma y tiempo los prácticos de máquina solicitados por la cátedra. Para aprobar la materia deberán rendir un examen final.

- Para promocionar la asignatura:

el alumno debe cumplir con las condiciones de regularización y aprobar los exámenes parciales con un nivel superior o igual al 70% del total y

resolver los ejercicios señalados particularmente en cada examen parcial (evaluación integradora) con un nivel superior o igual al 70% del total.

Se tomará una recuperación general adicional a los alumnos que hayan presentado certificado de trabajo en tiempo y forma.

EXAMEN LIBRE:

Consta de dos partes:

- 1) Desarrollo y entrega de prácticos de máquina.
- 2) Exámen teórico-práctico.

Los alumnos que entreguen y aprueben los prácticos de máquina (en tiempo y forma) podrán presentarse al exámen teórico-práctico.

IX - Bibliografía Básica

[1] The C Programming Language. Brian Kernighan - Dennis Ritchie. 1988. Prentice Hall. ISBN-10: 0131103628

[2] Apuntes de Lenguaje C (proporcionado por la Cátedra)

[3] Notas de clase sobre Recursión. (proporcionado por la Cátedra)

[4] Notas de clase sobre Programación Funcional. (proporcionado por la Cátedra)

[5] Haskell: The Craft of Functional Programming (2nd Edition), Simon Thompson, April 8, 1999, Addison-Wesley, ISBN 0-201-34275-8.

[6] A Discipline of Programming, Edsger W. Dijkstra, Prentice Hall, Inc. (October 28, 1976), ISBN-10: 013215871X, ISBN-13: 978-0132158718

[7] The Science of Programming (Monographs in Computer Science), David Gries, Springer (February 1, 1987), ISBN-10: 0387964800, ISBN-13: 978-0387964805

[8] Object-Oriented Programming: An Evolutionary Approach, Addison-Wesley Pub (Sd); 2 Sub edition (May 1991), ISBN-10: 0201548348, ISBN-13: 978-0201548341.

[9] Thinking in Java (4th Edition), Bruce Eckel, Prentice Hall; 4 edition (February 20, 2006), ISBN-10: 0131872486, ISBN-13: 978-0131872486.

[10] Sun Microsystems, The Java tutorial. A practical guide for programmers. Available online at <http://java.sun.com>

X - Bibliografía Complementaria

[1] C: A Software Engineering Approach. Peter A. Darnell, Philip E. Margolis. Springer; 3rd edition (May 29, 1996), ISBN-10: 0387946756

[2] Principles of Functional Programming. Hugh Glaser, Chris Hankin, David Till. Prentice Hall (November 1984), ISBN-10: 0137091486

[3] Flanagan D, Java in a Nutshell: A desktop quick reference (3rd Edition), O'Reilly, 1999.

XI - Resumen de Objetivos

Completar y reforzar los conceptos básicos de la programación imperativa.

Iniciar el estudio y aplicación del paradigma de la programación funcional

Iniciar el estudio y aplicación del paradigma de la programación orientada a objetos

XII - Resumen del Programa

1. El paradigma de la Programación Imperativa:

Lenguaje C: Tipos de datos simples y estructurados. Estructuras de control de flujo. Punteros. Administración dinámica de memoria. Entrada y Salida. Manejo de Archivos.

2. El paradigma de la Programación Funcional:

Lenguaje Haskell: Definición de funciones simples, recursivas, polimórficas y de orden superior. Manipulación de listas.

Funciones de plegado de listas (foldl, foldr).

3. El paradigma de la Programación Orientada a Objetos:

Lenguaje Java: Desarrollo de código Java básico en algún entorno (IDE) particular. Declaración , creación y destrucción de Objetos. Control de accesos. Implementación de Herencia.

XIII - Imprevistos

--

XIV - Otros

--

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
--	--

	Profesor Responsable
--	-----------------------------

Firma:	
--------	--

Aclaración:	
-------------	--

Fecha:	
--------	--