



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 2011)  
 (Programa en trámite de aprobación)  
 (Presentado el 08/06/2011 19:01:31)

### I - Oferta Académica

Materia	Carrera	Plan	Año	Período
PROGRAMACION II	LIC.CS.COMP.	32/12	2011	1° cuatrimestre
PROGRAMACION II	PROF.CS.COMPUT.	06/09	2011	1° cuatrimestre

### II - Equipo Docente

Docente	Función	Cargo	Dedicación
NECCO, CLAUDIA MONICA	Prof. Responsable	P.Adj Exc	40 Hs
SANCHEZ, ALBERTO ANTONIO	Responsable de Práctico	JTP Simp	10 Hs
VILALLONGA, GABRIEL DOMINGO	Responsable de Práctico	JTP TC	30 Hs
ALBORNOZ, MARIA CLAUDIA	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
0 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
16/03/2011	24/06/2011	15	135

### IV - Fundamentación

En este curso se refuerzan y completan los conocimientos y aptitudes que los alumnos adquirieron en Programación I, programando en el paradigma de la programación estructurada, y se los inicia en otros dos paradigmas: prog. Orientada a objetos y prog. Declarativa Funcional.

En esta asignatura se prepara al alumno para poder comparar las características de los distintos modelos formales subyacentes en el desarrollo de programas, utilizando lenguajes representativos de los tres paradigmas mencionados para introducir conceptos a partir del uso de los mismos. Esta práctica facilitará y/o servirá de base tanto para el análisis comparativo formal del desarrollo de lenguajes, como para el desarrollo de aplicaciones reales.

### 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 de orden superior, definición y aplicación.  
Currificación y aplicación parcial de funciones. Composición de funciones. Manejo de listas.  
Listas por comprensión.

### Lenguaje asociado: Haskell.

Entorno de trabajo, definición de programas, uso del intérprete.  
Módulos. Notación de listas [n..m]. Notación de listas por comprensión. 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.

### 3. Programación Orientada a Objetos

Descripción del paradigma de orientación a Objetos. Objetos. Mensajes. Clases.  
Conceptos: super, self. Encapsulamiento. Herencia. Control de herencia. Herencia multiple.  
Sobrecarga y Polimorfismo.

### Lenguaje asociado: C++. Características generales. Datos y funciones miembro.

Constructores y destructores. Sobrecarga de operadores. Control de acceso. Herencia.  
Visibilidad de métodos y datos. Compatibilidad de objetos. Funciones virtuales.  
Linking dinámico vs estático.

## VII - Plan de Trabajos Prácticos

Prácticos de aula:

1. Desarrollo de ejercicios en Lenguaje C
2. Desarrollo de ejercicios en Lenguaje Haskell
3. Desarrollo de ejercicios en Lenguaje C++

Prácticos de máquina:

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 exámen 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] Apuntes de Lenguaje C (proporcionado por la Cátedra)

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

[6] Aplique Turbo C++, Schildt, Herbert, McGraw-Hill, 1991, ISBN:84-7615-708-8.

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

## 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] Object-Oriented Programming: An Evolutionary Approach. Brad J. Cox - Andrew J. Novobilski. Addison-Wesley Pub (Sd); 2 Sub edition (May 1991), ISBN-10: 0201548348

[4] An Introduction to object Oriented Programming in C++. Wiener and Pinson. Addison-Wesley Professional (January 1, 1988), ISBN-10: 0201154137

## XI - Resumen de Objetivos

Completar y reforzar los conceptos básicos de la programación imperativa.

Iniciar el estudio 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:

Implementación de estructuras abstractas dinámicas. Punteros. Administración dinámica de memoria.

Entrada y Salida. Archivos. Lenguaje C

2. El paradigma de la Programación Funcional:

Características generales de los lenguajes funcionales.

Lenguaje de programación Funcional Haskell

3. El paradigma de la Programación Orientada a Objetos:

Descripción del paradigma de orientación a Objetos. Objetos. Mensajes. Clases. Herencia. Visibilidad. Compatibilidad de objetos. Lenguaje C++

## XIII - Imprevistos

**XIV - Otros**

--

<b>ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA</b>	
	<b>Profesor Responsable</b>
Firma:	
Aclaración:	
Fecha:	