



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 2008)
 (Programa en trámite de aprobación)
 (Presentado el 19/03/2008 11:20:29)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
PROGRAMACION II	LIC.EN CS.DE LA COMPUTACION		2008	1° cuatrimestre
PROGRAMACION II	PROF.EN CS.DE LA COMPUTACION		2008	1° cuatrimestre
PROGRAMACION II	TCO.UNIV.EN WEB		2008	1° cuatrimestre
PROGRAMACION II	TCO.UNIV.EN REDES DE COMP.		2008	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
NECCO, CLAUDIA MONICA	Prof. Responsable	P.Adj Exc	40 Hs
VILALLONGA, GABRIEL DOMINGO	Responsable de Práctico	JTP TC	30 Hs
ALBORNOZ, MARIA CLAUDIA	Auxiliar de Práctico	A.1ra Semi	20 Hs
PALACIO, GABRIELA DEL VALLE	Auxiliar de Práctico	A.2da Simp	10 Hs
PEREZ, NORMA BEATRIZ	Auxiliar de Práctico	A.1ra Semi	20 Hs
SANCHEZ, ALBERTO ANTONIO	Auxiliar de Práctico	JTP Simp	10 Hs
BRITOS MANRIQUE, LUIS EDUARDO	Auxiliar de Laboratorio	A.2da Simp	10 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
Hs	4 Hs	3 Hs	3 Hs	10 Hs

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

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
10/03/2008	20/06/2008	15	150

IV - Fundamentación

Cobertura de los contenidos necesarios descriptos en:

- Ciclo Obligatorio de la Licenciatura en Ciencias de la Computación, Anexo III, Ordenanza C.S. 11/98.
- Profesorado en Ciencias de la Computación, Anexo II, Ordenanza C.D. 003/00.
- Técnico Universitario en Web, Anexo II, Ordenanza C.D. 008/04.
- Técnico Universitario en Redes de Computadoras, Contenidos Mínimos, Anexo II, Ordenanza C.D. 007/04.

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:

Características generales del lenguaje C. Tipos numéricos. Operaciones. Conversiones de tipos. Casts. Control de secuencia. Arreglos. Estructuras. Uniones. Punteros. Relación entre punteros y arreglos. Aritmética de punteros. Strings. Entrada y Salida. Administración dinámica de memoria. Declaraciones de tipos. Clases de almacenamiento. Alcance. Archivos.

2. Programación Funcional:

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

Razonamiento formal. Transparencia referencial. Abstracción.

Lenguaje de programación Funcional de Backus: Definición. Objetos. Aplicación. Funciones Primitivas. Formas funcionales (Combinadores). Definición de funciones. Iteración. Recursión. Uso de acumuladores en funciones recursivas.

3. Programación Orientada a Objetos

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

Características generales del Lenguaje C++: Datos y funciones miembro. Constructores y destructores. Control de acceso. Encapsulamiento. Herencia. Control de herencia. Herencia multiple. Polimorfismo. Funciones virtuales. Objetos dinámicos.

VII - Plan de Trabajos Prácticos

Prácticos de aula:

1. Desarrollo de ejercicios en Lenguaje C
2. Desarrollo de ejercicios en Lenguaje Funcional de Backus
3. Desarrollo de ejercicios en Lenguaje C++

Prácticos de maquina:

Desarrollo y entrega de un práctico de cada uno de los paradigmas vistos: programación procedural, programación funcional y programación orientada a objetos.

VIII - Regimen de Aprobación

- Para regularizar la asignatura:

el alumno debe aprobar dos exámenes parciales o sus correspondientes recuperaciones, y presentar en forma y tiempo los prácticos de máquina solicitados por la cátedra.

- 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.

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

EXAMEN LIBRE:

Los alumnos que habiendo aprobado todos los prácticos de máquina hayan quedado libres por parciales, y en este caso, por la no aprobación de uno y no más de los temas que componen la materia, pueden presentarse como alumnos LIBRES. Una lista con los datos de los alumnos habilitados para rendir el examen libre será elevada a sección de alumnos junto con las

actas correspondientes a la finalización de la materia.

La posibilidad de rendir el examen en calidad de libre expira con el inicio del próximo dictado de la materia.

Los alumnos rendirán un examen final que contendrá ejercicios prácticos correspondientes al tema objeto de la pérdida de la condición de regularidad, más el contenido correspondiente a un examen final normal (el que debe rendir un alumno regular).

IX - Bibliografía Básica

[1] Tutorial de Lenguaje C (proporcionado por la Cátedra)

[2] Tutorial 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.

[6] Can programming be liberated from the Von Neumann Style?. John Backus. ACM Communications, vol.21, nro 8.

X - Bibliografía Complementaria

[1] C: A Software Engineering Approach. Peter A. Darnell, Philip E. Margolis. 1996, Springer Verlag.

[2] Aplique Turbo C++. Herbert Schildt. McGraw-Hill. 1991.

[3] Principles of Functional Programming. Hugh Glaser, Chris Hankin, David Till.

[4] Object-Oriented Programming: An Evolutionary Approach. Brad J. Cox - Andrew J. Novobilski. 1991. Addison

[5] Wesley.

[6] An Introduction to object Oriented Programming in C++. Wiener and Pinson.

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. Programación Imperativa.

Características generales del lenguaje C. Tipos de datos simples y estructurados. Punteros. Administración dinámica de memoria.

Entrada y Salida. Archivos.

2. El paradigma de la Programación Funcional

Características generales de los lenguajes funcionales.

Lenguaje de programación Funcional de Backus

3. El paradigma de la Programación Orientada a Objetos

Descripción del paradigma de orientación a Objetos. Objetos. Mensajes. Clases. Lenguaje C++

XIII - Imprevistos

XIV - Otros

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA

Profesor Responsable

Firma:

Aclaración:

Fecha: