Ministerio de Cultura y Educación **(Programa del año 2015)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
PROGRAMACION I	PROF.CS.COMPUT.	02/16	2015	2° cuatrimestre
PROGRAMACION I	LIC.CS.COMP.	32/12	2015	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
BAIGORRIA FERNANDEZ, LORENA S.	Prof. Responsable	P.Adj Exc	40 Hs
SILVESTRI, MARIO ALFREDO	Responsable de Práctico	JTP Simp	10 Hs
ALBORNOZ, MARIA CLAUDIA	Auxiliar de Práctico	A.1ra Semi	20 Hs
ESCUDERO, LEONARDO RUBEN	Auxiliar de Práctico	A.2da Simp	10 Hs
PEREZ, NORMA BEATRIZ	Auxiliar de Práctico	A.1ra Semi	20 Hs
VILLEGAS, MARIA PAULA	Auxiliar de Práctico	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	2 Hs	2 Hs	5 Hs	9 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
10/08/2015	20/11/2015	15	135

IV - Fundamentación

Se introduce al alumno en los conceptos fundamentales de la programación imperativa, acompañando la enseñanza de los conceptos teóricos con prácticas realizadas en un lenguaje de programación imperativo.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se espera que el alumno sea capaz de:
- Utilizar eficaz y eficientemente un enfoque de programación basado en el paradigma imperativo.
- Aplicar los principales conceptos del paradigma como por ejemplo los vinculados a las estructuras de control a nivel de sentencias y subprogramas, tipos de datos y control de datos.
- Diseñar, codificar y depurar programas de complejidad moderada utilizando un lenguaje representativo del paradigma.
- Diseñar y codificar soluciones recursivas usando un lenguaje imperativo.

VI - Contenidos

Unidad I

Paradigma de programación Imperativa. Concepto de lenguaje de programación. Introducción conceptual al Paradigma de Programación Imperativa utilizando el lenguaje C. Historia y características del lenguaje C. Fundamentos de C. Operadores y

expresiones. Tipos de Datos. Variables simples y estructuradas: registros y arreglos. Enfoque conceptual y aspectos operativos de las sentencias de control.

Unidad II
Programación estructurada. Estructuras de control básicas y compuestas. Programación estructurada utilizando C. Modularidad. Concepto de módulo. Definición y uso de funciones en C. Distintos tipos de funciones. Pasaje de parámetros. Parámetros formales y reales. Ambientes de referenciación.

Unidad III
Punteros. Naturaleza conceptual de un puntero. Conceptos de “apunta a” y “apuntado por”. Funcionamiento de punteros en C. Arreglos y asignaciones estáticas de memoria. Arreglos en C. Arreglos y punteros en C. Concepto de “cadena de caracteres”. Manejo conceptual de las cadenas de caracteres. Singularidades del manejo de cadenas de caracteres en C.

Unidad IV
Concepto de recursividad. Asignación de memoria (stack) durante la ejecución de un algoritmo recursivo. Posibilidad de desborde de la pila (stack overflow) durante la ejecución de algoritmos recursivos. Recursividad en C.

Unidad V
Concepto de “tipo de dato abstracto”. Ejemplo de “tipo de dato abstracto”. Manejo de tipos abstractos de datos en C. Listas como ejemplo de asignación dinámica de memoria. Asignación de memoria en el cúmulo (heap). Listas en lenguaje C. Modificaciones de listas. Pilas y sus aplicaciones. Colas y sus aplicaciones.

Unidad VI
Entrada/Salida. Entrada Salida Standard. Salida con formato. Argumentos de longitud variable. Entrada con formato. Manejo de Archivos. Manejo de errores. Descriptores de archivos. Operaciones de acceso a archivos. Acceso random. Operaciones de entrada salida de bajo nivel.

VII - Plan de Trabajos Prácticos

Los trabajos prácticos incluyen prácticas de aula y de laboratorio.
Trabajo Práctico I: Operadores y expresiones. Tipos de Datos. Variables simples. Estructuras de control básicas y compuestas.
Trabajo Práctico II: Modularidad. Definición y uso de funciones en C. Pasaje de parámetros. Punteros en C.
Trabajo Práctico III: Arreglos y punteros en C.
Trabajo Práctico IV: Registros en C.
Trabajo Práctico V: Tipos de datos abstractos
Trabajo Práctico VI: Recursividad.
Trabajo Práctico VII: Datos recursivos.
Trabajo Práctico VIII: Entrada/Salida.
Trabajo Práctico Integrador

VIII - Regimen de Aprobación

Condiciones para regularizar la materia:
Aprobar los prácticos de máquina y un examen parcial.
Contar con el 80% de asistencia a clases.

Prácticos de máquina: Entregar y aprobar el/los práctico/s de máquina. Un práctico de máquina se evalúa como aprobado o desaprobado, únicamente.

Exámenes parciales: Aprobar un examen parcial o sus respectivas recuperaciones, con nota mayor o igual que seis para su regularización.

Se otorga, tal como lo expresa la reglamentación vigente, dos recuperaciones por parcial a todos los alumnos.

Condiciones para promocionar la materia:
Regularizar la materia con las siguientes condiciones adicionales:
Aprobar el parcial o sus recuperatorios con nota mayor o igual que 7 (siete).
Rendir una evaluación integradora.

En caso de no promocionar el alumno deberá rendir un examen final escrito u oral.

Exámenes libres según lo dispuesto por Art 27 de Ord 13/03 CS.

IX - Bibliografía Básica

[1] Aristides Dasso & Ana Funes, Introducción a la Programación: Notas de clase para el curso Programación I, 2014.
[2] Kernighan & Ritchie, El Lenguaje de Programación C, 2da edición, PRENTICE-HALL, 1991, ISBN 9789688802052.
[3] Paul Deitel & Harvey Deitel, How to programa in C (6th edition), Prentice Hall, 2010

X - Bibliografia Complementaria

[1] Javier Fernández Muñoz, Problemas resueltos de programación en Lenguaje C, Ed. Paraninfo,2004, ISBN: 8497321022 ISBN-13: 9788497321020.
[2] Joyanes, Luis, Estructuras de datos en C. Serie Schaum, Mc Graw-Hill, 2005.
[3] Kruse, Robert Leroy; Tondon, Clovis L.; Leung, Bruce P: “Data structures and program design in C", Prentice-Hall,1997.
[4] Programación Estructurada. Dahl, Dijkstra, Hoare. Ed. Tiempo Contemporáneo.
[5] Algoritmos + Estructuras de Datos = Programas. N. Wirth. Ed. del Castillo.
[6] El Arte de la Programación, Vol I. D. Knuth. Ed. Reverte

XI - Resumen de Objetivos

Conocer los principales conceptos del paradigma de programación imperativa y desarrollar una experiencia de programación donde el alumno sea capaz de diseñar, codificar y depurar programas de complejidad moderada utilizando un lenguaje de programación representativo del paradigma imperativo.

XII - Resumen del Programa

Concepto de lenguaje de programación. Paradigma de Programación Imperativa. Operadores y expresiones. Variables. Estructuras de control básicas y compuestas. Tipos de Datos. Arreglos. Registros. Programación estructurada. Modularidad. Ambientes de referenciación. Punteros. Asignación estática y dinámica de la memoria. Tipo de dato abstracto. Listas, pilas y colas. Entrada/Salida. Recursividad.

XIII - Imprevistos

XIV - Otros