



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 2010)
(Programa en trámite de aprobación)
(Presentado el 21/08/2010 18:19:28)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
PROGRAMACION I	LIC.EN CS.DE LA COMPUTACION	006/05	2010	1° cuatrimestre
PROGRAMACION I	PROF.EN CS.DE LA COMPUTACION	06/09	2010	1° cuatrimestre
PROGRAMACION I	TCO.UNIV.EN WEB	008/05	2010	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
DASSO, ARISTIDES JUAN	Prof. Responsable	P.Tit. Exc	40 Hs
SILVESTRI, MARIO ALFREDO	Responsable de Práctico	JTP Simp	10 Hs

III - Características del Curso

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

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

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
15/03/2010	25/06/2010	15	135

IV - Fundamentación

Introducir al alumno en el paradigma de la programación imperativa, tal como los contenidos mínimos de los distintos planes de estudio de las carreras para las que se dicta la materia lo establecen, abordando todos los temas de este paradigma, tales como estructuras de control a nivel de sentencias, variables simples y estructuradas, tipos de datos predefinidos y definidos por el usuario y sus usos en resolución de problemas, registros, archivos, estructuras de control a nivel de unidades, parámetros formales y actuales, así como su aplicación en programas en un lenguaje que responda al paradigma, además de la práctica a través de laboratorios de implementación y ejecución de programas cortos permitirá al alumno desarrollar las habilidades necesarias para poder desarrollar programas en cualquier lenguaje de este paradigma.

V - Objetivos / Resultados de Aprendizaje

Se espera que el alumno adquiera el conocimiento de los elementos básicos del paradigma de la programación imperativa. Como tal deberá poder usar las estructuras de control a nivel de sentencias, variables simples y estructuradas, tipos de datos predefinidos y definidos por el usuario y sus usos en resolución de problemas.

En cuanto al manejo de estructuras de datos se pretende que pueda manejar las estructuras básicas tales como arreglos, registros, pilas, filas, listas.

Asimismo se espera que pueda utilizar las estructuras de control a nivel de unidades o módulos, así como el uso de

parámetros formales y actuales. Todo esto en programas en un lenguaje que responda al paradigma. También se espera que conozca y utilice como herramienta de resolución de problemas la recursión, tanto a nivel de algoritmos como de estructuras de datos recursivas.

VI - Contenidos

Pr-requisitos, Conceptos Previos

Lógica.

Introducción a la Lógica. Proposiciones. Funciones proposicionales. Variables proposicionales. Funtors de verdad (a) Conectivos: Conjunción Disjunción, Condicional, Bicondicional. (b) No-conectivos: Negación.

Las Computadoras

Las Computadoras. Las partes de una computadora. Las partes internas: Unidad Central de Proceso. Procesadores y Microprocesadores. Procesadores y Co-procesadores. Memoria Principal. Las partes externas. Teclado. Video. Memoria auxiliar: Discos magnéticos: Discos magnéticos flexibles (floppy ó disquete). Discos magnéticos rígidos (duros). La información (datos) dentro de la computadora. El funcionamiento. El Sistema Operativo.

Resolución de Problemas y Programación

Programación de computadoras. Programación y Resolución de Problemas. El Proceso de Resolución de Problemas. Etapas de la Resolución de Problemas. Los Paradigmas de la Programación. Programación Imperativa. Programación y Datos. Estructuras de programación. Lenguajes de programación: características generales. Métodos de descripción de lenguajes de programación. Sintaxis y semántica. Metalenguajes de definición.

Datos y Acciones

Acciones y descripción de datos. Datos: estándar, escalares. Definición y declaración. Constantes, variables, tipos. Acciones: Sentencias de asignación, iteración, selección, sentencias compuestas. Estructuras de Datos: Arreglos. Modularidad. Arreglos: orden de sus elementos, capacidad, operaciones. Modularidad: Subalgoritmos

Conceptos Nuevos

Unidad I

Introducción al Lenguaje de Programación Pascal. Orígenes, objetivos. Acciones y descripción de datos. Datos: estándar, escalares. Definición y declaración. Constantes, variables, tipos. Acciones: Sentencias de asignación, iteración, selección, sentencias compuestas. Estructuras de Datos. Orden de sus elementos: cronológico, no cronológico. Capacidad: dinámica, estática. Operaciones. Arreglos: orden de sus elementos, capacidad, operaciones. Registros: orden de sus elementos, capacidad, operaciones. Lenguaje de programación Pascal: tipos record y array.

Unidad II

Modularidad. Concepto de módulo. Lenguaje de programación Pascal: Procedimientos y funciones. Declaración, invocación, parámetros. Pasaje de parámetros. Resolución de problemas y uso de procedimientos y funciones.

Unidad III

Estructuras de datos I. Pilas: orden de sus elementos, capacidad, operaciones. Filas o colas: orden de sus elementos, capacidad, operaciones. Extensiones al lenguaje Pascal para manejo de estructuras dinámicas: pilas y filas. Declaraciones. Operaciones. Predicados.

Unidad IV

Estructuras de datos II. Listas uni-direccionales y bi-direccionales: composición de los elementos, orden de sus elementos, capacidad, operaciones. Generalidad de las listas. Extensiones al lenguaje Pascal para manejo de estructuras dinámicas: listas. Declaraciones. Operaciones. Predicados. Estructuras de más de un nivel: estructuras multinivel. Estructuras estáticas. Arreglos multinivel. Registros multinivel.

Unidad V

Implementación de Estructuras de Datos I. Conceptos Generales: Administración de los espacios libres. Desborde y desfonde. Métodos de administración de los espacios libres: administración estática, administración dinámica. Implementación de pilas: con y sin desplazamiento. Implementación de filas o colas: con y sin desplazamiento.

Unidad VI

Implementación de Estructuras de Datos II. Implementación de listas: con y sin desplazamiento.

Unidad VII

Recursividad. Conceptos Generales y aplicaciones. Procedimientos y funciones recursivas.

Unidad VIII

Tipo pointer: Conceptos Generales y aplicaciones.

Unidad IX

Datos recursivos. Conceptos Generales. Implementación de estructuras de datos, listas uni y bi-direccionales, empleando el tipo pointer.

Unidad X

Introducción al lenguaje C. Definiciones y declaraciones, datos y acciones. Estructuras de Datos: Array (arreglo) y struct (registro). Modularidad: funciones, parámetros.

VII - Plan de Trabajos Prácticos

1. En aula: PASCAL: definiciones y declaraciones, datos y acciones. Estructuras de Datos: PASCAL: Array (arreglo) y record (registro). En máquina: de los temas del práctico de aula.
2. En aula: Modularidad. PASCAL: declaraciones y uso de procedure y function. En máquina: de los temas del práctico de aula.
3. En aula: Estructuras de Datos: pilas, filas. Declaraciones y operaciones empleando las extensiones del PASCAL.
4. En aula: Estructuras de Datos: listas uni y bi-direccionales. Declaraciones y operaciones empleando las extensiones del PASCAL.
5. En aula: Implementación de estructuras: pilas, filas, empleando PASCAL. En máquina: de los temas del práctico de aula.
6. En aula: Implementación de estructuras: listas uni y bi-direccionales, empleando PASCAL. En máquina: de los temas del práctico de aula.
7. En aula: Recursividad. En máquina: de los temas del práctico de aula.
8. En aula: Tipo Pointer: PASCAL. En máquina: de los temas del práctico de aula.
9. En aula: Datos recursivos. Implementación de estructuras, listas uni y bi-direccionales, empleando el tipo pointer del PASCAL. En máquina: de los temas del práctico de aula.
10. En aula: Lenguaje C, definiciones y declaraciones, datos y acciones. Estructuras de Datos: Array (arreglo) y struct (registro). Tipo pointer. En máquina: de los temas del práctico de aula.

VIII - Regimen de Aprobación

1. Regularización y Aprobación.

La materia se desarrolla con dos modalidades: Promoción sin Examen Final (Artículos 34 y siguientes, Ord. 13/03CS) y Examen Final (Artículos 30 y siguientes, Ord. 13/03CS).

1.1. Examen Final

Para regularizar la materia, y poder rendir el correspondiente Examen Final, se deberá:

1.1.1. Tener como mínimo un 80% de asistencia a las clases de la asignatura.

1.1.2. Tener los prácticos, pedidos por la cátedra, aprobados. Son dos (2) Trabajos Prácticos a entregar.

1.1.3. Aprobar la Evaluación Final Integradora con un mínimo de 60%.

1.2. Promoción sin Examen Final:

Para regularizar y aprobar la materia se deberá:

1.2.1. Cumplir con los requisitos 1.1.1 y 1.1.2.

1.2.2. Aprobar la Evaluación Final Integradora con un mínimo de 85%. El porcentaje de 85 a 100 se corresponderá con una escala de siete (7) a diez (10), la que será la nota definitiva en la materia.

La evaluación continua de los alumnos se realizará a partir de los prácticos, tanto de aula como de laboratorio que los mismos realicen, así como de la evaluación, y aprobación, de los prácticos que los alumnos deban entregar. La Evaluación Final Integradora consistirá en la realización de un examen escrito cuyo objetivo es permitir evaluar la capacidad creativa de los alumnos y la integración global de los conocimientos adquiridos. En la nota final de ésta evaluación se tomará en cuenta los resultados obtenidos de la evaluación continua realizada a los alumnos.

2. Examen Final.

Aquellos alumnos que solo regularicen la materia podrán rendir el examen final, en los turnos establecidos.

Para rendir la materia en forma libre deberá rendirse un examen especial que consiste en dos partes: Práctica y Teórica. Para poder rendir la parte Teórica deberá aprobarse previamente la parte Práctica. La parte Práctica consistirá de uno o más ejercicios prácticos de programación, alguno de los cuales puede solicitarse que se implemente en máquina.

IX - Bibliografía Básica

[1] Pascal: Manual del Usuario e Informe. K. Jensen y N. Wirth. Ed. El Ateneo. 1984.

[2] El lenguaje de programación C. Brian W. Kernighan y Dennis M. Ritchie. Segunda Edición. Ed. Prentice Hall Hispanoamericana, Mexico, 1991.

[3] Manuales de los sistemas operativos empleados (MS-WINDOWS, Linux).

[4] "Notas de Clase" de la cátedra.

X - Bibliografía Complementaria

[1] Programación Estructurada. Dahl, Dijkstra, Hoare. Ed. Tiempo Contemporáneo. 1977.

[2] Algoritmos + Estructuras de Datos = Programas. N. Wirth. Ed. Dossat. 2a. edición 1999.

[3] El arte de programar ordenadores: Algoritmos fundamentales. Editorial Reverté. 1986.

XI - Resumen de Objetivos

Adquirir el conocimiento teórico y práctico de los elementos básicos del paradigma de la programación imperativa.

XII - Resumen del Programa

Introducción a un Lenguaje de Programación imperativo.

Acciones y descripción de datos.

Datos: estándar, tipos.

Acciones: Sentencias de asignación, iteración, selección, sentencias compuestas.

Estructuras de Datos estáticas. Arreglos. Registros.

Modularidad.

Estructuras de datos dinámicas. Pilas. Filas. Listas.

Implementación de Estructuras de Datos dinámicas.

Recursividad.

Tipo pointer: Conceptos Generales y aplicaciones.

Datos recursivos.

XIII - Imprevistos

--

XIV - Otros

--

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	