Ministerio de Cultura y Educación **(Programa del año 2020)**
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	ING. EN COMPUT.	28/12 026/1	2020	1° cuatrimestre
PROGRAMACION I	ING. INFORM.	2-08/15	2020	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
BAIGORRIA FERNANDEZ, LORENA S.	Prof. Responsable	P.Adj Exc	40 Hs
PEREZ, NORMA BEATRIZ	Responsable de Práctico	JTP Exc	40 Hs
SANCHEZ, HECTOR ENRIQUE	Auxiliar de Práctico	A.1ra 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	2 Hs	6 Hs

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

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
09/03/2020	10/07/2020	18	108

IV - Fundamentación

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.

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. 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.

VII - Plan de Trabajos Prácticos

Las clases presenciales se vieron interrumpidas por el COVID-19 y el aislamiento social determinado por el gobierno nacional, es por esto que los estudiantes realizaron sus prácticas en sus domicilios tanto en papel como en computadora.

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 Integrador de Laboratorio realizado en forma individual.

VIII - Regimen de Aprobación

Para realizar una evaluación continua de los estudiantes se realizan practicas de laboratorio de 40 minutos en tiempo real. Un práctico de laboratorio se evalúa como aprobado o desaprobado. En caso de ser aprobado se considera como puntaje adicional para la evaluación parcial.

Condiciones para regularizar la materia:

- Aprobar el examen parcial con nota mayor o igual a 6 (seis).
- Contar con el 70% de las actividades de seguimiento presentadas a través del aula virtual.
- Aprobar el laboratorio integrador con nota mayor o igual a 6 (seis)

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 examen parcial con nota mayor o igual a 7 (siete).
- Contar con el 80% de las actividades de seguimiento presentadas a través del aula virtual.
- Aprobar el laboratorio integrador con nota mayor o igual a 7 (siete)
- Aprobar un coloquio oral por videoconferencia con nota mayor o igual que 7 (siete)

En caso de no promocionar el alumno deberá rendir un examen final escrito u oral.
No se admiten exámenes libres debido al carácter eminentemente práctico de la materia.

IX - Bibliografía Básica

[1] El Lenguaje de Programación C; Kernighan & Ritchie; Ed. Pearson; 1991.
[2] How to program in C (8th edition); Paul Deitel & Harvey Deitel; Ed. Prentice Hall; 2016.
[3] Data structures and program design in C; Kruse, Robert; Ed. Pearson; 2007.
[4] Algoritmos + Estructuras de Datos = Programas; N. Wirth; Ed. Del Castillo; 1999.
[5] Programacion en C; Serie Schaum, Byron S. Gottfried; Ed. Mcgraw-Hill / Interamericana De España; 2005.
[6] Fundamentos de Informatica y Programacion en C; Diego Rafael Llanos Ferraris ; Ed. ParaInfo; 2010.
[7] Aristides Dasso & Ana Funes, Introducción a la Programación: Notas de clase para el curso Programación I, 2014.
[8] Videos explicativos realizados por la cátedra

X - Bibliografia Complementaria

[1] Estructuras de datos en C; Serie Schaum, Joyanes, Luis; Ed. Mc Graw-Hill; 2005.
[2] Problemas resueltos de programación en Lenguaje C; Javier Fernández Muñoz; Ed. Parainfo; 2004.
[3] Programación Estructurada. Dahl, Dijkstra, Hoare. Ed. Tiempo Contemporáneo.
[4] 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

Las clases presenciales fueron suspendidas por el aislamiento social decretado por el gobierno, continuando el dictado de la materia de manera no presencial. Esto llevó a la modificación del cronograma original, cambiando el ritmo del dictado ya que el estudiante requirió mas tiempo para la asimilación de los contenidos debido al carácter eminentemente practico de la

materia. Debido a esto se extendió el dictado de la materia.
Las clases teóricas y prácticas se realizan en forma no presencial a través de videoconferencias. Se realiza un seguimiento continuo de las actividades prácticas a través del aula virtual ademas del sitio de la materia. Así mismo se realizan videos tanto teóricos como prácticos como soporte de la explicación de los contenidos de la materia. En forma complementaria se hace uso de grupo de whatsapp creado para la materia como soporte de comunicación para los estudiantes.
El parcial se dividió en dos partes por cuestiones de duración y conectividad:
Teoría: la misma se realiza a través de un cuestionario en el aula virtual de la materia, en tiempo acotado, en horario establecido.
Práctica: la misma se toma en tiempo real, por videoconferencia y las resoluciones de puño y letra del estudiante son subidas en forma inmediata en el aula virtual.
El laboratorio integrador debe ser subido en el aula virtual tanto código fuente como ejecutable para ser probado y luego defendido por el estudiante.
Las evaluaciones han sido acordadas y comunicadas a los estudiantes en el aula virtual y en el sitio de la materia.

Aulas virtuales de la FCFMyN, UNSL
https://www.evirtual.unsl.edu.ar/moodle/

Sitio web:
http://proguno.unsl.edu.ar

Contacto:
programacion1.unsl@gmail.com
proguno@unsl.edu.ar

Docentes de la materia:
Oficina 1, 2 y 3 del dpto de informática.
1er piso, bloque 2.

XIV - Otros