

FORMULARIO DE PRESENTACIÓN DE CURSOS DE POSGRADO

1.1. Indique la denominación del curso propuesto:

Programación en C.

1.2. Inserto en un carrera de posgrado

Sí No

1.3. En caso de que el curso ya sea dictado en otra carrera indique la siguiente información:

Carrera	Tipo de dictado	Modalidad	Carácter
Lic. en Ciencias Básicas de la FCEN	Semanal	presencial y/o virtual debido a la situación sanitaria	electivo

2. Equipo docente.

2.1. Responsable a cargo.

Apellido: Millán
Nombre: Emmanuel Nicolas
Documento: DNI 29974940
Correo electrónico: emmanuel.millan@itu.uncu.edu.ar
CUIT/CUIL: 20-29974940-2

2.2. Integrantes del equipo docente (repetir cuantas veces sea necesario)

Apellido:.....
Nombre:.....
Documento:.....
Correo electrónico:.....
CUIT/CUIL:.....

3. Fecha probable de dictado

Semestre 1er 2do mes: Agosto/Septiembre

4. Número máximo y mínimo de alumnos

Máximo 10

5. Carga horaria propuesta:

100 horas (60 horas de cursado teórico-práctico y 40 horas de elaboración de un proyecto integrador)

5.1. Exprese la carga horaria relacionada al dictado de la actividad en horas reloj.

Modalidad	Carga teórica	Carga práctica	Total	Porcentaje
Presencial	20	40	60	60
No presencial	0	40	40	40
Total	20	80	100	

Debido a la situación sanitaria la totalidad de la carga horaria se realizará a distancia.

6. Objetivos

1. Dominar los conceptos básicos del lenguaje C
2. Identificar y organizar los datos pertinentes al problema, evaluando el contexto particular.
3. Comprender las ventajas y desventajas del lenguaje de C.
4. Conocer las fuentes de información disponibles para que el/la estudiante pueda seguir mejorando sus habilidades.
5. Aplicar el lenguaje para resolver problemas relacionados a la especialidad del estudiante.
6. Ser capaz de relacionar el conocimiento profundo de las ciencias básicas con la resolución de problemas a través del lenguaje de programación C.
7. Aplicar el pensamiento lógico, científico y sistémico sobre problemas de ciencias básicas.

7. Contenidos. (2000 caracteres)

Unidad 1: Fundamentos de lenguajes de programación

Lenguajes de programación, traductores y compiladores. La compilación y sus fases. Paradigmas de programación (imperativos, declarativos, orientados a objetos). Metodología de programación (modular, orientada a objetos, estructurada). Diseño de algoritmos, escritura de algoritmos. Diagramas de flujo. Pseudocódigo. Herramientas de programación: IDE, compilador, depurador, make/cmake.

Unidad 2: Introducción a C

Estructura general. Creación de un programa básico. Depuración (errores de sintaxis, lógicos, regresión). Mensajes de error, errores en tiempo de ejecución. Elementos de un programa de C. Tipos de datos C. Variables, alcance. Entrada y salida básica. Operadores. Preprocesador. Formatos de entrada y salida de flotantes, enteros, cadenas.

Unidad 3: Estructuras de selección y control:

Condicionales if y switch. Bucles while, for, do while. Bucles infinitos. Break y continue

Unidad 4: Funciones:

Estructura de una función. Prototipos de las funciones. Parámetros por valor y referencia. Ámbitos: de una variable, del programa, archivo fuente, función y bloque. Funciones de bibliotecas. Funciones numéricas, matemáticas, trigonométricas, generación de números aleatorios. Funciones recursivas.

Unidad 5: Arreglos, caracteres y cadenas:

Declaración, ejemplos, índices. Arreglos multidimensionales. Pasaje de arreglos a funciones. Concepto de cadena de caracteres. Lectura de cadenas. Biblioteca string.h. Funciones varias para el tratamiento de cadenas

Unidad 6: Punteros:

Direcciones de memoria. Concepto de puntero. Operadores para punteros. Llamada a funciones por referencia. Relación entre punteros y arreglos. Arreglos de apuntadores. Asignación de memoria dinámica

Unidad 7: Estructuras y uniones:

Definición y uso de estructuras. Estructuras anidadas. Arreglos a estructuras. Uniones

Unidad 8: Procesamiento de Archivos:

Flujos, apuntador a archivo. Apertura y cierre de un archivo. Funciones de entrada/salida para archivos. Archivos binarios. Funciones para acceso aleatorio. Datos externos al programa (argumentos de main)

Unidad 9: Introducción a algoritmos y estructuras de datos avanzadas:

Listas enlazadas, pilas, colas y árboles. Algoritmos de ordenación y búsqueda. Conceptos básicos de arquitectura de computadoras. Ejemplo de paralelización básica con OpenMP. Pruebas de rendimiento.

8. Describa las actividades prácticas desarrolladas, indicando lugar donde se desarrollan y modalidad de supervisión. (Si corresponde). (2000 caracteres)

Las actividades prácticas se llevarán a cabo durante el horario de clase programado, en conjunto con las secciones teóricas. Actualmente se prevé que el trabajo práctico representará aproximadamente el 60% de las clases programadas, aunque este número puede variar de una clase a otra.

Las actividades prácticas tomarán la forma de ejercicios guiados y orientados hacia las ciencias básicas. Como las actividades prácticas se llevarán a cabo durante la clase y utilizando los mismos materiales (es decir, computadoras, lápiz y papel), se llevarán a cabo en el mismo espacio que la clase teórica. Por lo tanto, no se requerirá supervisión adicional de los asistentes. En caso de ser necesario por la situación sanitaria, el espacio curricular está preparado para ser dictado 100% virtual

9. Bibliografía propuesta

1. Joyanes Aguilar, L., & Zahonero Martínez, I. (2014). Programación en C, C++, JAVA y U ML. México DF: Mc. Graw-Hill. Segunda Edición
2. Mathieu, M. J. (2014). Introducción a la programación. Grupo Editorial Patria.
3. Kernighan, B. W., & Ritchie, D. M. (1991). *El lenguaje de programación C*. Pearson Educación.
4. Deitel, H. M., & Deitel, P. J. (2004). *Cómo programar en C/C++ y Java*. Pearson educación.
5. Perry, G. M., & Miller, D. (2014). *C programming: absolute beginner's guide*. Pearson Education.
6. Shaw, Z. A. (2015). *Learn C the hard way: practical exercises on the computational subjects you keep avoiding (like C)*. Addison-Wesley Professional.
7. Seacord, R. C. (2020). *Effective C: An Introduction to Professional C Programming*. No Starch Press.
8. Prata, S. (2014). *C primer plus*. Pearson Education.
9. Griffiths, D., & Griffiths, D. (2012). *Head First C: A Brain-Friendly Guide*. O'Reilly Media, Inc..

10. Modalidad de evaluación y requisitos de aprobación y promoción.

Los asistentes tendrán una evaluación integradora al final del cursado y deberán entregar un trabajo final integrador. La evaluación integradora tomará la forma de ejercicios computacionales, en los cuales los asistentes deberán crear o completar programas informáticos en C para resolver problemas específicos. El acceso al material del curso e internet no se restringirá durante las evaluaciones, ya que refleja un entorno de trabajo realista. Los ejercicios enviados se evaluarán en función de su capacidad para resolver el problema, la eficiencia del código, la legibilidad del código y la calidad de salida (cuando corresponda).

El trabajo final de investigación se enfocará en la utilización de técnicas avanzadas de programación en C, por lo que se requerirá un desarrollo mayor y el uso de técnicas más complejas de programación para la elaboración de dicho trabajo final. El tema del trabajo final podrá ser alineado con los temas de investigación del estudiante de posgrado, por ejemplo, vinculado a su tesis. Los asistentes también serán evaluados en su desarrollo durante los aspectos prácticos de las clases.

La evaluación del estudiante será de la siguiente manera:

La calificación final será un promedio de las notas obtenidas en la evaluación integradora y el trabajo final. Las notas de la evaluación integradora y el trabajo final deberían ser iguales o mayores a 6 (seis) para aprobar.

Modalidad de examen para estudiantes libres

- ❖ Presentar todos los trabajos prácticos realizados durante el cursado.
- ❖ Aprobar una evaluación práctica al finalizar el cursado que abarca todo el material del curso.

- ❖ Aprobar una evaluación oral al finalizar el cursado para determinar el nivel de comprensión del material del curso.
 - La asignatura se considerará aprobada cuando se aprueben las dos evaluaciones con una nota igual o superior a 7 (siete).
- ❖ Presentar un proyecto equivalente al trabajo final de la materia.

Las notas obtenidas en cada uno de los tres puntos detallados anteriormente deberá ser igual o superior a 7 (siete). La nota final será un promedio de las tres calificaciones obtenidas.

11. Ingrese toda otra información que considere pertinente, incluidos requisitos específicos si corresponde.

1. Para que el curso sea efectivo, el profesor requerirá el uso de la sala de informática seis horas por semana, divididas en dos clases de tres horas.
2. Debido a los recursos limitados en el punto uno (1) habrá un cupo máximo de asistentes. Se dará prioridad a los solicitantes de FCEN-UNCuyo, seguidos por los estudiantes vocacionales de otras facultades de UNCuyo y, finalmente, de los solicitantes que no estén asociados con UNCuyo.
3. Para enseñar el curso de manera efectiva, el profesor requerirá el uso de un proyector digital, y que una distribución de Linux tenga instalado el compilador GCC y un editor de textos en cada una de las máquinas en la sala de computación.