



CÓDIGO DE ASIGNATURA
0612
1110

DEPARTAMENTO: *Ingeniería e Investigaciones Tecnológicas*

ASIGNATURA: *Programación II - Programación* Ingeniería en Informática
Año: 2do. Cuatri: 1ro.

1. OBJETIVOS

Marco de referencia

La Universidad debe formar ingenieros con capacidad creadora, actitud crítica y disposición a la actualización permanente, así como con una destacada formación teórica y experimental.

Una sólida formación en las materias de ciencias básicas y en las tecnologías básicas (este último es el caso de esta materia), asegurará que el futuro ingeniero se adapte a la diversidad de situaciones que se presenten en su desempeño profesional.

Ante el acelerado avance de la tecnología y los consecuentes requerimientos de un cada vez más alto grado de especialización, el dominio de tales tecnologías básicas es un recurso indispensable para poder interpretar, aprehender y utilizar los nuevos conocimientos y desarrollos tecnológicos, para así asumir los cambios y participar en la creación de nuevos conocimientos y desarrollos tecnológicos.

En este contexto, la Programación, constituye una herramienta tecnológica fundamental que permite la creación de modelos de hechos y / o situaciones de entidades diversa, desde aspectos de la naturaleza, hasta el desarrollo de sistemas de gestión de información, pasando por la simulación de procesos que de otro modo requerirían el uso intensivo y a veces destructivo de elementos reales puestos a prueba; sin descontar su importancia para el aprendizaje de asignaturas posteriores de la carrera tales como Sistemas Operativos, Lenguajes y Compiladores, Análisis y Diseño de Sistemas, Bases de Datos, Redes de Comunicaciones, Seguridad Informática, etc..

La misma tiene además un buen grado de vinculación con materias de años anteriores de otras áreas (Cálculo y Discreta) y de asignaturas del mismo año.

Dado lo anterior, la materia se desarrollará teniendo en cuenta los siguientes objetivos:

- Que el alumno logre desarrollar algoritmos para la resolución de diversas situaciones, utilizando un lenguaje de programación (el Lenguaje C en nuestro caso).



- **Que pueda expresar los contenidos teóricos de la materia y su vinculación con situaciones de la vida real (manipulación de archivos con información binaria o de texto, trabajar con arrays uni y bidimensionales, simulación de pilas y colas de espera, listas y árboles binarios).**
- **Que implemente algoritmos relacionados con temas de otras materias del curso anterior y del mismo año.**
- **Que comprenda y aplique los principios de la Programación Orientada a Objetos, utilizando un lenguaje de programación (el Lenguaje C++ en nuestro caso), y comprenda que un Objeto es una entidad capaz de contener un estado interno (información privada) y un comportamiento (que se manifiesta por los métodos de la clase), los conceptos de herencia simple y múltiple, el polimorfismo, la implementación de mensajes entre objetos de distintas clases, y el tratamiento de las situaciones de excepción. Todo esto a partir de que tales objetos son creados instanciándolos a partir de clases.**
- **Que el alumno vea la fuerte vinculación de la POO con materias de años posteriores tales como Construcción de Sistemas e incluso Ingeniería de Software.**
- **Que el alumno se sienta estimulado a integrarse en grupos de trabajo, potenciando su propio aprendizaje a través de la interacción y cooperación con sus pares.**
- **Que adquiera, como un valor agregado, el conocimiento de tales herramientas de programación (Lenguaje C y Lenguaje C++), que le serán necesarios en asignaturas posteriores.**

Para la realización de los trabajos prácticos se han elegido dos IDEs (Entornos de Desarrollo Integrado), la herramienta Visual Studio Visual C / C++ de Microsoft, debido a la gran facilidad que brinda esta herramienta para la ejecución paso a paso, depuración de errores, modificaciones del código o del contenido de variables en tiempo de ejecución, y puesta a punto de proyectos integrados por varios programas fuente. Los alumnos que utilicen esta herramienta en sus trabajos prácticos, recibirán una certificación emitida por Microsoft y por la Cátedra de Programación II. De lo contrario el alumno podrá utilizar a su gusto cualquier otra herramienta de desarrollo como la distribución gratuita del IDE CodeBlocks y su compilador MinGW (Minimalist GNU for Windows).

Además, en esta materia (1110 Programación), se implementa un Trabajo de Calificación Profesional, que en la materia se implementa en un Taller de Programación, con una carga horaria de 32 horas, brindando la posibilidad de que los alumnos elijan entre un Taller de Programación en Lenguaje Java, y agregando desde el año pasado, por indicación de las autoridades, otros dos Talleres de Programación en .net-C# y en Delphi, a fin de brindar la mencionada posibilidad de elección por parte de los alumnos. En los mismos se hará una introducción a su uso en que los alumnos avanzarán desde la realización de programas simples a modo consola, avanzando rápidamente al uso



de los recursos de ventanas y su consiguiente Programación Orientada a Eventos. En el mismo se darán los conceptos de POO incorporando la Programación Orientada a Eventos y su uso en programas interactivos con el usuario empleando cuadros de diálogo, botones, menús, etc.. Se los guiará en el desarrollo de diversas aplicaciones interactivas con objetivos y alcances diferenciados según la plataforma elegida. Los alumnos que lo aprueben satisfactoriamente obtendrán una certificación de su aprobación emitida por Oracle, Microsoft o Embarcadero según el taller elegido y la Cátedra de Programación II (aún en trámite los acuerdos con los dos primeros, ya acordada con Embarcadero).

2. CONTENIDOS

Conceptos y elementos básicos del Lenguaje C. Sentencias de control utilizadas en C. Implementación y uso de las funciones en C. Recursividad. Arrays y cadenas de caracteres. Arreglos de caracteres y archivos de texto. Punteros. Tipos definidos por el usuario. Manejo de archivos en C. Archivos de Datos. Funciones standard de búsqueda y ordenamiento. Estructuras de datos. Tipos de Datos Recursivos. Representación de datos en memoria. Estrategias de implementación. Manejo de memoria en ejecución. Estructuras dinámicas en C. Diseño y Análisis de algoritmos. Tipos de Datos Abstractos Pila, Cola,, Lista y Árbol, variantes usos y aplicaciones. Paradigma de Programación orientada a objetos. Conceptos fundamentales. Programación Orientada a Objetos utilizando C++. Clases, Herencia y Polimorfismo. Encapsulamiento y Abstracción. Introducción a la programación en JAVA. Incluye TCP (Trabajo de Calificación Profesional), el que para esta materia se implementa en un Taller de Programación en Java o .net-C# o Delphi a elección de los alumnos.

Unidad 1 Arrays y archivos

Introducción a la abstracción de datos y ocultamiento de información. Funciones de búsqueda, conversión, ordenamiento, etc.. Manejo de arrays con notación y aritmética de punteros. Recursividad. Funciones de las bibliotecas string.h, ctype.h, math.h, stdlib.h, stdio.h, etc.. Arrays bidimensionales. Arrays de punteros. Punteros a funciones. Funciones de biblioteca qsort, bsearch y lsearch. Archivos binarios y de texto, su creación, modos de acceso, posicionamiento, cierre, eliminación, y funciones relacionadas. Creación de tipos de datos, uniones, enumeraciones, macroemplazos. Generación de arrays en tiempo de ejecución. Argumentos variables.

Unidad 2 TDA o Estructura de datos Pila

Asignación dinámica de memoria vs. asignación estática de memoria. Primitivas para el manejo de Pilas, compatibilidad de primitivas entre la implementación estática y dinámica de Pilas. Su relación con la recursividad.

Unidad 3 TDA o Estructura de datos Cola como un TDA

Asignación dinámica de memoria vs. asignación estática de memoria. Pri-



mitivas para el manejo de Colas, compatibilidad entre la implementación estática y dinámica de Colas. Su relación con el "bufer" de teclado.

Unidad 4 TDA o Estructura de datos Lista como un TDA

Primitivas para el manejo de Listas. Creación, inserción, ordenamiento, búsqueda, eliminación, etc., con asignación dinámica de memoria.

Listas circulares, su importancia en la implementación de colas y su parecido y diferencia con la implementación de pilas.

Listas doblemente enlazadas.

Unidad 5 TDA o Estructura de datos Árbol

Árbol binario, creación, recorridas EnOrden, PreOrden y PosOrden. Árbol binario de búsqueda, su relación con la búsqueda binaria en arrays. Determinación de altura, y otras funciones.

Árboles AVL y balanceados, determinación.

Unidad 6 Introducción a C++ – Programación Orientada a Objetos (POO)

Modificaciones menores con respecto al Lenguaje C. Comentarios, punteros constantes, sobrecarga de funciones, parámetros por defecto, parámetros por alias (referencia). Asignación y liberación dinámica de memoria, operadores new y delete. Flujos de entrada/salida, manipuladores.

Programación Estructurada frente a la POO. Arquitectura y evolución del diseño de software. Clases, objetos, métodos y mensajes. Lenguajes que soportan la POO y su evolución.

Unidad 7 Programación Orientada a Objetos con C++

Clases, objetos, métodos de la clase (funciones miembro), funciones amigas. Especificación inline y const. Constructores y destructores, su ejecución automática. Puntero this. Sobrecarga de operadores. Arrays, punteros y objetos.

Unidad 8 Herencia

Conceptos de herencia, y su por qué. Introducción a herencia simple y herencia múltiple. Clases derivadas. Diagramas de Jerarquía. Control de acceso a la clase base. Especificador protected.

Unidad 9 Excepciones

Manejo de excepciones, throw, catch, try. Especificación de excepciones. Funciones estándar assert, terminate y unexpected.

Unidad 10 Introducción al Lenguaje Java

Escritura, compilación y ejecución de programas sencillos desde la línea de comando. Uso de un Entorno de Desarrollo Integrado para la generación de aplicaciones de consola e interactivas. Tipos de datos, operadores, precedencia de operadores, clases predefinidas en el lenguaje (String y otras).

4. BIBLIOGRAFÍA

Título	Autor(es)	Editorial	Año Edición	Ejemplares disponibles en UNLaM
El Lenguaje de Programación C	Kernighan y Ritchie	Prentice Hall		



UNIVERSIDAD NACIONAL DE LA MATANZA

Título	Autor(es)	Editorial	Año Edición	Ejemplares disponibles en UNLaM
<i>El Lenguaje de Programación C++</i>	<i>Bjarne Stroustrup</i>	<i>Addison Wesley</i>		
<i>Cómo Programar en C / C++</i>	<i>Deitel y Deitel</i>	<i>Prentice Hall</i>		
<i>C / C++, Manual de Referencia</i>	<i>Herbert Schildt</i>	<i>Mc Graw Hill</i>		
<i>Cómo Programar en C / C++ / Java</i>	<i>Deitel y Deitel</i>	<i>Prentice Hall</i>		
<i>Data Structures and Program Design in C</i>	<i>Kruse, Leung y Tondo</i>	<i>Prentice Hall</i>		
<i>Thinking in C++</i>	<i>Bruce Eckell</i>	<i>Prentice Hall</i>		
<i>Estructuras de Datos con C y C++</i>	<i>Langsam, Augenstein y Tenembaum</i>	<i>Prentice Hall</i>		
<i>Programación Orientada a objetos</i>	<i>Luis Joyanes Aguilar</i>	<i>Mc Graw Hill</i>		
<i>Apuntes de la cátedra</i>				
<i>Para la Introducción a Java.</i>				
<i>Apuntes, Tutoriales y ayudas provistos por la cátedra</i>				
<i>Tutorial de Java:</i>	<i>http://download.oracle.com/javase/tutorial/</i>	<i>Sun (hoy Oracle)</i>		
<i>Cómo Programar en Java</i>	<i>Deitel y Deitel</i>	<i>Prentice Hall</i>		
<i>Thinking in Java</i>	<i>Bruce Eckell</i>	<i>Prentice Hall</i>		
<i>Páginas de tutoriales</i>	<i>NetBeans y Sun</i>	<i>NetBeans y Sun (hoy Oracle)</i>		