UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
INGENIERÍA EN INFORMÁTICA
PLAN 2008
PROGRAMA DE ESTUDIO
ANEXO 01

I. IDENTIFICACIÓN

1.	Asignatura	: Lenguajes de Programación I
2.	Semestre	: Segundo
3.	Horas semanales	: 7 horas
	3.1. Clases teóricas	: 3 horas
	3.2. Clases prácticas	: 4 horas
4.	Total real de horas disponibles	: 112 horas
	4.1. Clases teóricas	: 48 horas
	4.2. Clases prácticas	: 64 horas

II. JUSTIFICACIÓN

Esta materia está orientada a formar al profesional informático centrándose en el enfoque de la comprensión de las necesidades en el campo de la programación. Esta área es fundamental para el profesional y de su correcto aprendizaje que los sistemas informáticos construidos cumplan sus objetivos específicos en forma eficaz y eficiente. Se pretende entregar al alumno un conjunto de conocimientos que sean balanceados y flexibles de manera tal que le permitan aplicar en su ejercicio profesional desde que se entienda que este puede estar inserto en ambientes bastantes disímiles y contemplando además un campo laboral tanto local como regional que se caracteriza por la competitividad. Al efecto se ha elaborado un plan de estudios que comprende las herramientas iniciales para introducir al alumno al mundo de la programación, dando un enfoque general de los conceptos y técnicas fundamentales de la programación estructurada con una de las herramientas más modernas, exitosamente utilizado para el área de enseñanza como de desarrollo comercial y científico como es el lenguaje C. Con lo anterior se dotará al alumno con un lenguaje que extiende su continuidad en C++/Java/Rubi/C# entre otros modernos lenguajes y su intrínseca capacidad en aplicaciones tanto de gestión como científicas. El lenguaje C es un lenguaje de medio-alto nivel por lo que dará capacidades de abstracción al alumno para posteriormente abordar de forma más simple otros lenguajes comerciales de alto nivel.

III. OBJETIVOS GENERALES

Comprender los fundamentos y aplicar los conceptos del paradigma de la programación estructurada y modular en el diseño de soluciones a problemas de la vida real utilizando como herramienta base el Lenguaje C como un primer contacto a los distintos enfoques de programación y herramientas de desarrollo de software.

IV. OBJETIVOS ESPECÍFICOS

A. Conocimientos

1. Describe claramente los conceptos de programación estructurada y modular.
2. Comprende la estructura lógica o pasos en el proceso de diseño de un código ejecutable de alto nivel a partir de un algoritmo de solución bajo el compilador de C.
3. Conoce las técnicas de las buenas prácticas de programación para el desarrollo de códigos claros y simples.
4. Conoce las bondades, limitaciones, terminología y la sintaxis del lenguaje C.
5. Clasifica correctamente los tipos de datos y tipos de funciones desde la perspectiva de lenguaje C.
6. Comprende los conceptos de memoria local, externa, entradas y salidas y sus implicancias aplicativas.

B. Habilidades

1. Aplica correctamente los conceptos de programación estructurada y modular en el diseño de soluciones basada en software.
2. Apreciar el papel central que juega la abstracción en la tarea de programador.
3. Desarrolla una aproximación disciplinada a la especificación, implementación y documentación de programas.
4. Utiliza correctamente las herramientas de ambiente de desarrollo integrado para la edición, corrección y puesta a punto de código en versión fuente y ejecutable.
5. Aplica correctamente los tipos y estructuras de datos y funciones para el desarrollo de un código escalable y portable.
6. Diseña correctamente interfaces básicas para entrada y salida de datos tanto en interfaces estándares como la memoria local y externa a través de archivos.

C. Competencias

1. Capacidad de aplicar los conocimientos en la práctica.
2. Disposición para el trabajo en equipo.
3. Capacidad para identificar, plantear y resolver problemas básicos
4. Capacidad de abstracción y análisis para el diseño de algoritmos.
5. Capacidad en la búsqueda de nuevas informaciones y conocimientos tecnológicos para su aplicación.

V. PRE - REQUISITO

1. Matemática Discreta
2. Algoritmos y Estructuras de Datos I

VI. CONTENIDO

6.1. Unidades programáticas

1. Introducción.
2. Sintaxis del lenguaje C.
3. Estructuras de control de programa.
4. Arreglos multidimensionales y cadenas.
5. Punteros y manejo de memoria interna.
6. Funciones y procedimientos.
7. Manejo de caracteres y cadenas.
8. Datos básicos y agregados.
9. Tipo Abstracto de datos.
10. Estructuras de datos.
11. Manejo de programas por consola.
12. Manejo de memoria externa.
13. Compilación condicionada.

6.2. Desarrollo de las unidades programáticas

1. Introducción
1.1. Breve historia de los lenguajes de programación
1.2. Algoritmia
1. 3. Historia del C
1. 4 .C como lenguaje de nivel medio y estructurado.
1.5. Compiladores e intérpretes
1.6. Partes de un programa en C
1. 7. Bibliotecas
1.8. Compilación de un programa en C

2. Sintaxis del lenguaje C.
2.1. Los cinco tipos de datos básicos
2.2. Modificaciones de identificadores
2.3. Nombres de identificadores
2.4. Variables
2.5. Modificadores del tipo de acceso
2.6. Especificadores de clase de almacenamiento
2.7. Inicializaciones de variables
2.8. Constantes
2.9. Operadores
2.10. Expresiones

3. Estructuras de control de programa.
3.1. Verdadero y falso en lenguaje C
3.2. Sentencias de:
- 3.2.1. Iteración
- 3.2.2. Salto
- 3.2.3. Expresión
- 3.2.4. Bloque

4. Arreglos y cadenas.
4.1. Arreglos unidimensionales
4.2. Generación de un puntero a un arreglo
4.3. Paso de arreglo uni-dimensionales a funciones
4.4. Cadenas
4.5. Arreglos bidimensionales y multidimensionales a funciones
4.6. Indexación de punteros
4.7. Inicialización de arreglos

5. Punteros a memoria interna
5.1. Introducción a los punteros
5.2. Variables punteros
5.3. Los operadores de punteros
5.4. Expresiones de punteros

5.5. Punteros y arreglos
5.6. In-dirección múltiple
5.7. Inicialización de punteros
5.8. Punteros a funciones
5.9. Funciones de asignación dinámica de C

6. Funciones.
6.1. Forma general de una función.
6.2. Reglas de ámbito de las funciones
6.3. Argumentos de funciones
6.4. La sentencia return
6.5. Funciones que devuelven valores no enteros
6.6. Prototipos de funciones
6.7. Devolución de punteros
6.8. Funciones del tipo void
6.9. Lo que devuelve main
6.10. Recursividad
6.11. Declaración de listas de parámetros de longitud variable
6.12. Declaración de parámetros de funciones clásicas frente a las modernas
6.13. Aspectos de implementación
6.14. Bibliotecas y archivos

7. Manejo de caracteres y cadenas.
7.1. Fundamentos de cadenas y caracteres
7.2. Funciones de entrada y salida
7.3. Procesamiento de cadenas

8. Tipo abstracto de datos
7.4. El papel de la abstracción
7.5. Abstracción en los lenguajes de programación
7.6. Tipo de datos
7.7. Tipo abstracto de datos TAD
7.8. Creación de un TAD
7.9. Relación de TAD y objetos

9. Datos básicos y agregados
9.1. Estructuras
9.2. Arreglos de estructuras
9.3. Paso de estructuras a funciones
9.4. Punteros a estructuras
9.5. Arreglos y estructuras dentro de estructuras
9.6. Campo de bits, uniones y enumeraciones

10. Estructuras de datos
10.1. Estructuras lineales estáticas y dinámicas
10.1.1. Colas
10.1.2. Pilas
10.1.3. Listas enlazadas
10.2.Estructuras no lineales estáticas y dinámicas
10.2.1. Árboles
10.2.2. Grafos

11. Manejo de programas por consola
11.1.Escritura y lectura de caracteres
11.2.Escritura y lectura de cadenas
11.3.E/S por consola con formato

12. Manejo de memoria externa
12.1.Entrada y Salida por archivos.
12.2.Archivos Secuencias
12.3.Archivos Aleatorios
12.4.Elementos básicos del sistema de archivos

13. Compilación condicionada

13.1. El pre-procesador de C.
13.2. Directivas de compilación condicional.
13.3. Los operadores del pre-procesador # y ##.
13.4. Nombres de macros predefinidos.
13.5. Comentarios.

VII. ESTRATEGIAS METODOLÓGICAS

1. Actividades de dinámica de grupos para la facilitar la integración grupal, facilitar la producción individual y la concertación de soluciones grupales.
2. Resolución de problemas con la construcción de aplicaciones informáticas utilizando un software libre.
3. Propuesta de temas de investigación y ejercicios para desarrollo extra clase.
4. Concurso de programación basadas en la olimpiadas universitarias ACM/ICPC.
5. Cursos en eventos educacionales y tecnológicos (ETyC, ERTIC)

VIII. MEDIOS AUXILIARES

1. Pizarras acrílicas.
2. Marcadores.
3. Borrador de pizarra acrílica.
4. Computadoras.
5. Proyectores multimedia.
6. Parlantes para multimedia.
7. Plataforma virtual "EDUCA".
8. Sala de laboratorio equipada para las prácticas.
   - 8.1. Computadoras en red.
   - 8.2. Sistemas operativos Linux, Windows.
   - 8.3. Acceso a internet.

IX. EVALUACIÓN

Para evaluar la asignatura se tienen en cuenta lo siguiente:

1. Examen final compuesto por un test escrito y trabajo practico final.
2. Evaluación continua de teoría y práctica, obtenida con la media de los controles realizados durante el curso a través de test semanales y ejercicios de laboratorio computados en los exámenes parciales.
3. Las calificaciones se basan en el reglamento de la Universidad.

X. BIBLIOGRAFÍA

- Como programar en C/C+/Java. Deitel y Deitel. Prentice Hall. Ultima edición.
- Estructura de Datos una perspectiva en C. McGraw Hill, Aguilar y Martínez. Ultima edición.
- Kerninghan, C. B. W. El Lenguaje de Programación / C. B. W. Kerninghan, D.M. Ritchie. Prentice Hall, 1991.
- Alcalde, Eduardo. Metodología de la Programación. Eduardo Alcalde, Miguel García. - - Editorial MC Graw Hill. Ultima edición.
- Joyanes Aguilar, Luis. Fundamentos de Programación / Luis Joyanes Aguilar. - - Editorial Mc Graw Hill. Ultima edición.
- C: Guía de Auto enseñanza. H. Schildt McGraw-Hill, 1994.
- C: Manual de Referencia. H. Schildt. McGraw-Hill, 1989.

XI. ENLACES WEB

- www.cconclase.net
- www.cpluplus.com
-