

Planeación del curso Fundamentos de Programación

Trimestre 2016-I

1. Información General

Clave: 2151103

Grupo: CB02

Horario de clases teóricas: Martes de 10:00 a 12:00 (B303) y Viernes de 10:00 a 11:00 (E214)

Horario de prácticas: Lunes de 11:00 a 14:00 hrs. (AT-105)

Horario de asesorías: Lunes a Jueves de 15:00 a 16:00

Profesor: Graciela Román Alonso

email: grac@xanum.uam.mx

Oficina: T-215

2. Objetivos

General

Que al final del curso el alumno sea capaz de resolver problemas sencillos a través del desarrollo de programas haciendo uso de buenas prácticas de programación.

Específicos

Al finalizar el curso el alumno será capaz de:

- Plantear soluciones algorítmicas para problemas simples utilizando estructuras de control básicas.
- Utilizar adecuadamente los tipos de datos simples y estructurados.
- Traducir sus algoritmos en programas modulares bien estructurados.

3. Contenido Sintético

- | | | |
|-------|--------------------------------------|-------------|
| 1 | Conceptos básicos | (1 semana) |
| 1.1 | Modelo de Von Neuman | |
| 1.2 | Software básico | |
| 1.2.1 | Sistema operativo | |
| 1.2.2 | Lenguajes | |
| 1.2.3 | Editores de texto | |
| 1.2.4 | Compiladores | |
| 1.2.5 | Depuradores simbólicos | |
| 1.2.6 | Ambientes de Desarrollo integrado | |
| 1.3 | Algoritmo: definición y expresión | |
| 2 | Elementos de programación imperativa | (4 semanas) |

- 2.1 Datos
 - 2.1.1 Identificadores
 - 2.1.2 Constantes y variables
 - 2.1.3 Tipo de datos: definición
- 2.2 Tipos de datos simples
 - 2.2.1 Enteros
 - 2.2.2 Reales
 - 2.2.3 Caracteres
 - 2.2.4 Booleanos
- 2.3 Sentencias simples
 - 2.3.1 Asignación
 - 2.3.2 Entrada y salida
- 2.4 Construcción de expresiones aritméticas y lógicas
 - 2.4.1 Operadores aritméticos y su precedencia
 - 2.4.2 Operadores lógicos, tablas de verdad y precedencia
- 2.5 Estructuras de control
 - 2.5.1 Secuenciación
 - 2.5.2 Selección condicional simple, doble y múltiple
 - 2.5.3 Iteración condicional
- 2.6 Tipos de datos compuestos
 - 2.6.1 Principio de los tipos de datos compuestos
 - 2.6.2 Ejemplo: Cadenas
- 3 Estructura de un programa (2 semanas)
 - 3.1 Diseño de programación ascendente y descendente
 - 3.2 Módulo principal y subordinados
 - 3.3 Tipos de módulos: procedimientos y funciones
 - 3.4 Parámetros
 - 3.4.1 Definición
 - 3.4.2 Parámetros formales
 - 3.4.3 Parámetros actuales
 - 3.4.4 Tipos de paso de parámetros
 - 3.4.4.1 Valor
 - 3.4.4.2 Referencia
 - 3.5 Anidamiento de módulos
 - 3.6 Alcance de identificadores
- 4 Estructuras de datos básicas (2 Semanas)
 - 4.1 Arreglos unidimensionales
 - 4.2 Arreglos multidimensionales
 - 4.3 Registros
- 5 Documentación y prueba de programas (2 Semanas)
 - 5.1 Estándar de codificación
 - 5.2 Especificación de casos de pruebas

4. Bibliografía

- Allen, M., (2000), Estructuras de datos en java, Prentice Hall, EUA.
- Baldwin, D. Scragg, G.W., (2004), Algorithms and Data Structures: The Science of Computing, Cengage Charles River Media, EUA.
- Dale, N., Joyce, D. T., Weems, C., (2003), Object oriented data structures using java, Jones and Barlet Publishers, EUA.
- Guardati Bueno, S., (2007), Estructuras de datos orientadas a objetos: Algoritmos en C++, Pearson, México
- Joyanes Aguilar, L., (2003), Fundamentos de programación: algoritmos de datos y objetos; McGraw-Hill; España
- López R., L., (2003); Programación estructurada: un enfoque algorítmico; Algaomega; México.
- Sahni, S., (2005), Data Structures, Algorithms and Application with Java, Silicon Press, EUA.

5. Modalidad de conducción

- Exposición oral de los temas frente al grupo por parte del profesor.
- Discusión y solución de problemas de programación en clase.
- Sesiones de laboratorio para la puesta en práctica de los temas expuestos.

6. Evaluación del curso

- Evaluaciones parciales (3 exámenes) 50%
 - Examen 1 : viernes semana 4, 12 de febrero
 - Examen 2 : martes semana 8, 8 de marzo
 - Examen 3 : martes semana 11 BIS, 5 de abril
- Participaciones 15%
- Prácticas 20%
- Programas y Tareas 15%

Para aprobar el curso el promedio de exámenes deberá ser aprobatorio

7. Escala

8.6 <= MB <= 10

7.6 <= B < 8.6

6.0 <= S < 7.6

0.0 <= NA < 6.0