

NOMBRE DE LA INSTITUCIÓN	Universidad de Sonora
UNIDAD ACADÉMICA	Unidad Regional Centro
DIVISION ACADÉMICA	División Ciencias Exactas y Naturales
DEPARTAMENTO ACADÉMICO QUE IMPARTE SERVICIO	Departamento de Matemáticas
LICENCIATURAS USUARIAS	Licenciatura en Ciencias de la Computación
NOMBRE DE LA MATERIA	Programación de Computadoras
CLAVE	9444
EJE FORMATIVO	Básico
REQUISITOS	Álgebra superior I, Diseño de algoritmos
CARÁCTER	Obligatorio
VALOR EN CRÉDITOS	8 (2 teoría/2 lab. / 2 taller)
CLAVE	

Introducción

El rápido desarrollo de las tecnologías de la información tiene un impacto profundo en la sociedad actual. De aquí la importancia que tiene la computación y en particular el análisis, diseño y construcción de algoritmos bien estructurados y eficientes, así como las estructuras de datos utilizadas y la forma de pensar para conseguir soluciones a problemas de manera clara, eficaz y fácil de implementar en una computadora.

Este es un curso introductorio que ofrece los fundamentos de proceso y la metodología de la programación que inicia con un enfoque estructurado y presenta los conceptos básicos de la programación orientada a objetos utilizando el lenguaje C++. Los principios abordados en este contenido permitirán desarrollar las habilidades lógico-matemáticas de los alumnos en la creación de software, que madurarán en el transcurso de sus estudios profesionales.

Objetivo general

El alumno será capaz de comprender y aplicar las técnicas básicas para el desarrollo de soluciones a problemas elementales de procesamiento de información implementando la solución en el lenguaje de programación C++.

Objetivos particulares

El alumno será capaz de:

- Reconocer los principios básicos que sustentan el desarrollo de software profesional.
- Aplicar el proceso de desarrollo de software para la elaboración de los programas que resuelvan los problemas que se planteen en el curso.
- Enunciar soluciones en pseudocódigo.
- Conocer y aplicar las técnicas que permitan la solución de un problema de procesamiento de información usando la computadora.
- Estudiar y ejercitar la semántica y la sintaxis del lenguaje de programación C++.
- Destacar las principales estructuras de control de la programación en C++.
- Comprender y manejar los conceptos básicos de la programación orientada a objetos.

Contenido

1. Introducción al lenguaje de programación C++

- 1.1. Elementos de un programa en C++
 - 1.1.1. Tipos de datos
 - 1.1.2. Identificadores, constantes y variables
 - 1.1.3. Salida libre y con formato
 - 1.1.4. Entrada de datos
 - 1.1.5. Archivos de cabecera
- 1.2. Estructura básica de un programa en C++

- 1.3. Operadores
 - 1.3.1. Aritméticos, relacionales, lógicos y el operador *sizeof*
 - 1.3.2. Prioridad y asociatividad de operadores
- 1.4. Conversión de tipos implícita y explícita
- 1.5. Sentencias nula y compuesta
- 1.6. *typedef*, enumeraciones y el tipo de dato *bool*
- 1.7. Tipos de datos definidos por el usuario
 - 1.7.1. Arreglos unidimensionales y bidimensionales indexados
 - 1.7.2. Registros
- 2. Estructuras de control**
 - 2.1. Selección simple, doble y múltiple
 - 2.2. Operador condicional
 - 2.3. Estructuras de repetición
 - 2.3.1. Contadores, acumuladores y operadores
 - 2.3.2. Ciclos *while*, *for* y *do – while*
 - 2.4. Ciclos anidados
 - 2.5. Variantes de las estructuras de repetición
 - 2.5.1. Sentencias *break* y *continue*
- 3. Funciones**
 - 3.1. Definición, declaración y llamada a una función
 - 3.2. Punteros y los operadores *&* y ***
 - 3.3. Parámetros
 - 3.4. Punteros a funciones
 - 3.5. Funciones en línea
 - 3.6. Ámbito de funciones
 - 3.7. Sobrecarga de funciones
 - 3.8. Recursividad
- 4. Arreglos**
 - 4.1. Unidimensionales
 - 4.1.1. Punteros y arreglos unidimensionales
 - 4.1.2. Paso de arreglos unidimensionales a funciones
 - 4.1.3. Asignación dinámica de memoria
 - 4.1.4. Cadenas de caracteres y arreglos de caracteres
 - 4.2. Bidimensionales
 - 4.2.1. Punteros y arreglos bidimensionales
 - 4.2.2. Paso de arreglos unidimensionales a funciones
 - 4.2.3. Asignación dinámica de memoria
- 5. Registros**
 - 5.1. Punteros y estructuras
 - 5.2. Estructuras anidadas
 - 5.3. Arreglos y estructuras
 - 5.4. Estructuras y funciones
 - 5.5. Uniones
- 6. Introducción a las clases**
 - 6.1. Definición
 - 6.2. Acceso a sus miembros (encapsulamiento)
 - 6.3. Funciones miembro
 - 6.4. Archivos de cabecera y de clases
 - 6.5. Constructores
 - 6.6. Destruyores
 - 6.7. Excepciones

6.8. Sobrecarga de operadores

7. Archivos

7.1. Operaciones sobre archivos de texto

7.2. Manejo de la impresora

Estrategias Didácticas

- Promover la participación activa de los estudiantes en todas las sesiones del curso fomentando la discusión de los conceptos elementales que sustentan a la programación de computadoras y motivando el interés en la investigación en los distintos paradigmas de programación.
- Promover la consulta bibliográfica y en línea de los distintos temas que se abordarán en el curso, así como la presentación de informes y exposiciones orales.
- Para cada una de las unidades, plantear problemas típicos del tema en exposición y desarrollar la solución de éste a través de un proceso de desarrollo de software sistemático.
- Asegurarse del dominio de los temas, a través de la práctica en el desarrollo de algoritmos que solucionen diversos problemas planteados por el profesor en las sesiones de clase y, en particular, en tareas extra-clase.
- Diseñar prácticas de laboratorio apropiadas para cada unidad con la finalidad de que el alumno demuestre el dominio de los conceptos abordados en clase.
- Comparar las distintas soluciones que puede tener un problema, de acuerdo al trabajo realizado por el grupo.
- Motivar discusiones sobre eficiencia de algoritmos.
- Fomentar la elaboración de algoritmos en pseudocódigo y C++.

Estrategias de Evaluación

Para la evaluación de los estudiantes, se sugiere que el profesor tome en cuenta:

- Resultados de los exámenes parciales aplicados (se sugiere que sean al menos tres).
- Tareas, trabajos de investigación, presentaciones.
- Participación individual y colectiva en las actividades del curso.
- Elaboración de prácticas de laboratorio.
- Elaboración de un proyecto final.

Los porcentajes de cada indicador serán establecidos al inicio del semestre.

Bibliografía

- Deitel, H. M. y Deitel, P. J. *C++ Cómo programar*. 2da. ed., Prentice Hall. México, 1999.
- Bronson, Gary J. *C++ para ingeniería y ciencias*. International Thomson Editores. México, 2000.
- Strouptrop, Bjarne. *El lenguaje de programación C++*. 3ra. ed., Addison Wesley. Madrid, 1998.
- Joyanes A., L. *Programación en C++. Algoritmos, estructuras de datos y objetos*. McGraw Hill.
- Rodríguez A. E. y Rodríguez C. I. *Seudocódigo*. Material Didáctico No. 5. Depto. de Matemáticas, Universidad de Sonora, 2002.

Perfil académico deseable del profesor

Se recomienda que el profesor tenga el siguiente perfil:

- Formación profesional en el área de ciencias de la computación o área afín.
- Conocimiento y aplicación de los principios de desarrollo de software, en particular los paradigmas de diseño estructurado descendente y el orientado a objetos.