

Taller de
Programación de Computadoras con el Lenguaje C++
Aplicaciones didácticas y prácticas en matemáticas, ingeniería y ciencias

Ing. Arturo J. Miguel de Priego Paz Soldán
www.tourdigital.net
Chincha, Perú, mayo de 2015

C++ es un lenguaje de programación de computadoras que se utiliza ampliamente en la industria del software porque ofrece muchas facilidades para crear programas de alta escala siguiendo el paradigma orientado a objetos, la programación dirigida por funciones y la programación genérica. Este lenguaje de propósito general sirve para programar aplicaciones en ingeniería, ciencias, juegos, bases de datos, comunicaciones en redes de computadoras, procesamiento de imágenes, vídeo y música, control de procesos industriales, herramientas de diseño y enseñanza, publicidad, realidad virtual, etc.

En este taller los participantes adquirirán los conocimientos fundamentales y desarrollarán las técnicas básicas para adquirir, procesar, crear, almacenar y recuperar datos e información usando el lenguaje C++ en los sistemas operativos DOS y Windows. A lo largo del taller, los estudiantes aprenderán los fundamentos de la programación orientada a objetos utilizando la sintaxis y semántica de C++; conocerán la estructura y funcionamiento de las computadoras, de los microprocesadores y de las memorias; entenderán como se carga y ejecuta un programa; practicarán el proceso de desarrollo de software y desarrollarán proyectos de programación siguiendo un proceso de diseño en ingeniería.

Al concluir satisfactoriamente este taller, cada estudiante tendrá la capacidad de plantear y desarrollar soluciones de programación orientada a objetos, programación estructurada y paramétrica usando el lenguaje C++, y además será capaz de identificar oportunidades de programación y desarrollar soluciones metódicas, creativas y efectivas, especialmente en aplicaciones didácticas y prácticas en las disciplinas de matemáticas, ciencias e ingeniería. Algunos de los programas estudiados durante las clases son:

Matemáticas Entrenador para la tabla de multiplicar, juego de memoria, intersección de figuras geométricas, trazado de funciones trigonométricas; cálculo de media, mediana y moda; solución de ecuaciones de segundo y tercer grado; solución de un sistema de ecuaciones lineales de dos variables; un juego para adivinar un número de cuatro dígitos, etc.

Ciencias Vectores, distancia alcanzada por un proyectil, experimentos virtuales de cinemática, campos eléctricos, etc.

Ingeniería Contador con visualizador de siete segmentos, ley de Ohm, tutorial de puertas lógicas, simulador de un semáforo, controlador de un semáforo, controlador de intensidad de luz por un led con la técnica PWM, pasamensajes por un LCD, etc.

Cada tema se complementa con ejemplos y ejercicios de programación ilustrativos y en todo momento se procura que los estudiantes completen diagramas de bloques, pseudocódigos, y diagramas de flujos antes de codificar y depurar los programas.

OBJETIVOS INSTRUCCIONALES

Al finalizar el taller, los estudiantes tendrán la capacidad para:

1. Describir las unidades principales de un computador y explicar cómo el procesador ejecuta las instrucciones de un programa almacenado en la memoria.
2. Utilizar la sintaxis y la semántica del lenguaje C++ junto con las unidades básicas de control (secuencia, selección, iteración) y estructuras de programación (subprogramas, clases) para expresar ideas (cantidades, procesos, algoritmos) que resuelvan problemas de programación.
3. Identificar y seleccionar los algoritmos más efectivos para tareas comunes y básicas de programación en matemáticas, ciencias e ingeniería en los sistemas operativos DOS y Windows.
4. Organizar y almacenar la información utilizando estructuras de datos (listas, tablas, registros).
5. Reconocer procesos comunes de programación y patrones de estructuras de datos, de las estructuras de control y de los algoritmos que se aplican.
6. Realizar comunicaciones de datos mediante el teclado, el ratón, el monitor y archivos.
7. Interpretar y analizar el código de un programa C++, predecir el resultado de la ejecución, y ampliar y adaptar su desempeño.
8. Reconocer, plantear y llevar a cabo un proyecto de programación.
9. Comunicar efectivamente, mediante artículos y exposiciones, los resultados de un proyecto de programación a un público general y a una audiencia especializada.
10. Crear programas de computadora para tareas de moderada complejidad aplicando el diseño modular de programas, habilidades básicas para la solución de problemas y una aproximación sistemática prestando cuidadosa atención a cada fase del ciclo de vida del software.

CRONOGRAMA

1. Arquitectura y funcionamiento de una computadora personal. Representaciones de datos. Programas y compiladores. Ejecución de programas en DOS y en Windows.
2. Sintaxis y gramática de C++. Programación modular de tareas. Sentencias secuenciales, selectivas e iterativas. Solución de ecuaciones. Dibujo de funciones matemáticas.
3. Arreglos y punteros. Arreglos de datos. Búsqueda de datos. Juego de tres en raya. Punteros a datos. Memoria dinámica.
4. Biblioteca API de Windows. Controles. Interfaz gráfica. Eventos del teclado y ratón.
5. Programa para resolver suma de dos y tres vectores.
6. Programación orientada a objetos. Clases, objetos, constructores y destructores.
7. Herencia. Clases bases y clases derivadas. Programación de un tutorial de figuras geométricas. Combinación de clases.
8. Polimorfismo. Funciones virtuales. Programación de un tutorial de circuitos lógicos.
9. Animaciones. Creación de temporizadores. Dibujo de imágenes y creación de animaciones.
10. *Primera evaluación del desempeño académico.*
11. Estructura de datos. Listas simples y enlazadas. Tablas, diccionarios, mapas.
12. Archivos. Creación, edición, almacenamiento y recuperación de archivos.
13. La biblioteca STL. Programación genérica. Contenedores, iteradores y algoritmos.
14. Ejemplo de proyecto: un tutorial de circuitos lógicos.
15. Oportunidades para un proyecto de programación.
16. Diseños de los programas.
17. Verificación e integración de los programas y optimización del proyecto.
18. Presentación del proyecto.
19. *Segunda evaluación del desempeño académico.*
20. Reflexiones sobre los aprendizajes.