**Objetivos de la materia:**

* Conocer las estructuras de programación, las estructuras de datos y los operadores necesarios para la construcción de programas en C/C++.
* Adquirir la capacidad de evaluar alternativas y resolver situaciones problemáticas utilizando la programación como herramienta.
* Adquirir la capacidad de representar algoritmos mediante el uso del lenguaje C/C++.
* Adquirir nociones generales sobre el paradigma de programación estructurada.
* Adquirir nociones generales sobre el paradigma de programación orientada a objetos.
* Integrarse a grupos de trabajo.
* Evidenciar habilidad en el uso de vocabulario técnico específico.
* Adquirir nociones generales para el desarrollo de programas funcionales en C/C++.

| UNIDAD TEMÁTICA Nº: 1 |
| --- |

***Título: Estructuras de datos complejas en C/C++***

*Contenidos:*

Variables avanzadas.

Vida de una variable. Conversiones de tipo y casting.

Arreglos. Cadenas. Punteros. Concepto. Operadores de punteros (\*, &). Expresiones de punteros. Punteros y arreglos. Punteros y cadenas.

Registros: definición de tipo de dato struct. Uso. Acceso a los campos.

**Competencias específicas asociadas a la Unidad**

1. Adquirir conocimientos teóricos angulares sobre variables complejas.

4. Adquirir conocimientos lógico-algorítmicos para aplicar a la práctica de la materia

práctica de la materia

12. Aplicar fundamentos matemáticos y estadísticos en la construcción de soluciones informáticas para la práctica de la materia

| UNIDAD TEMÁTICA Nº: 2 |
| --- |

***Título: Funciones***

*Contenidos:*

Forma general de una función. Ámbito de las funciones. Variables globales y locales. Argumentos y parámetros de funciones: paso de parámetros por valor y dirección.

Prototipos de funciones. Tips de valores de retorno Funciones y objetos de datos . Funciones globales y de clase.

Sobrecarga de funciones y parámetros por omisión.

**Competencias específicas asociadas a la Unidad**

1. Adquirir conocimientos teóricos angulares acerca de las funciones en C/C++.

4. Adquirir conocimientos lógico-algorítmicos para aplicar a la práctica de la materia

9. Aplicar los conocimientos teóricos adquiridos en la construcción de programas.

11. Aplicar fundamentos lógico-algorítmicos en el desarrollo de programas para la práctica de la materia

12. Aplicar fundamentos matemáticos y estadísticos en la construcción de soluciones informáticas para la práctica de la materia

| UNIDAD TEMÁTICA Nº: 3 |
| --- |

***Título: Archivos***

*Contenidos:*

Accesos a disco. Archivos de texto y binarios. Entrada/Salida a nivel del sistema.

Archivos. Concepto. Trabajo sobre archivos. La estructura FILE. Apertura y cierre de un archivo. Escritura de un archivo. Lectura de un archivo. Búsqueda y modificación. Operaciones con archivos. Altas, modificaciones, consultas. Baja lógica de elementos de un archivo binario. Copia y comparación de archivos.

**Competencias específicas asociadas a la Unidad**

1. Adquirir conocimientos teóricos angulares acerca de la definición, creación y gestión de archivos.

4. Adquirir conocimientos lógico-algorítmicos para aplicar a la práctica de la materia

9. Aplicar los conocimientos teóricos adquiridos en la construcción de programas.

11. Aplicar fundamentos lógico-algorítmicos en el desarrollo de programas para la práctica de la materia

12. Aplicar fundamentos matemáticos y estadísticos en la construcción de soluciones informáticas para la práctica de la materia

| UNIDAD TEMÁTICA Nº: 4 |
| --- |

***Título: Asignación dinámica de memoria***

*Contenidos:*

Gestión dinámica de memoria. Estructuras dinámicas de datos: colas, pilas y listas enlazadas. Insertar y eliminar elementos. Generación, recorrido y destrucción de una estructura dinámica de datos.

**Competencias específicas asociadas a la Unidad**

1. Adquirir conocimientos teóricos angulares sobre gestión dinámica de memoria

9. Aplicar los conocimientos teóricos adquiridos en la construcción de programas.

| UNIDAD TEMÁTICA Nº: 5 |
| --- |

***Título: Introducción a la Programación Orientada a Objetos (POO) en C++***

*Contenidos:*

Diferencias entre Programación Estructurada y Programación Orientada a Objetos. Definición de Clases y Objetos. Diseño de clases. Especificadores de acceso. Conceptos de abstracción y encapsulamiento. Constructores y destructores.

**Competencias específicas asociadas a la Unidad**

1. Adquirir conocimientos teóricos angulares acerca de los fundamentos de la POO.

4. Adquirir conocimientos lógico-algorítmicos para aplicar a la práctica de la materia

8. Aplicar el enfoque sistémico y pensamiento divergente en el análisis y resolución de problemas

9. Aplicar los conocimientos teóricos adquiridos en la construcción de programas.

11. Aplicar fundamentos lógico-algorítmicos en el desarrollo de programas para la práctica de la materia

| UNIDAD TEMÁTICA Nº: 6 |
| --- |

***Título: Conceptos avanzados de la Programación Orientada a Objetos (POO) en C++***

*Contenidos:*

Herencia y polimorfismo. Sobrecarga de operadores y funciones.

Nociones de diseño de sistema. Definición de clases y archivos y módulos. Determinación de las funciones de la clase y funciones globales.

Codificación, compilación, ejecución y depuración de programas ejemplo. Desarrollo de un programa funcional bajo el paradigma de la POO.

**Competencias específicas asociadas a la Unidad**

1. Adquirir conocimientos teóricos angulares acerca de los conceptos centrales de la POO.

4. Adquirir conocimientos lógico-algorítmicos para aplicar a la práctica de la materia

8. Aplicar el enfoque sistémico y pensamiento divergente en el análisis y resolución de problemas

9. Aplicar los conocimientos teóricos adquiridos en la construcción de programas.

11. Aplicar fundamentos lógico-algorítmicos en el desarrollo de programas para la práctica de la materia

12. Aplicar fundamentos matemáticos y estadísticos en la construcción de soluciones informáticas para la práctica de la materia

15. Resolver situaciones problemáticas planteadas como proyectos interdisciplinares integradores

**Competencias actitudinales asociadas a la materia:**

5.- Adquirir el vocabulario y la terminología propia de la materia

7.- Adquirir hábitos metodológicos y organizados de estudio y práctica

13. Aplicar los resultados del autoaprendizaje, actualización e investigación en la resolución de problemas.

17.- Realizar las actividades prácticas en forma continua

18.- Presentar las actividades prácticas en tiempo y forma (presentación de soluciones y terminología)

19.- Desempeñar las actividades dentro del equipo con responsabilidad individual, grupal, ética y social, en tiempo y forma

20.- Participar activamente en el aula (virtual o presencial), realizando aportes a la comunidad de aprendizaje

21-Adquirir una lectura comprensiva de situaciones problemáticas

**Bibliografía obligatoria para el alumno:**

* Zang, T (2000). *Aprendiendo C en 24 hs*. México: Prentice Hall.
* Perry, G. (1995). *Aprendiendo Programación Orientada a Objetos con TURBO C++ en 21 Días.* Madrid, España: Prentice Hall.
* Morero, F (1999). *Introducción a la OOP.* Versión 1.0.0 – Madrid, España: Grupo EIDOS.
* Materiales de la cátedra

**Bibliografía para el docente:**

* Deitel, H y Deitel, P (2000). *Cómo programar en C/C++*. México: Pearson Education.
* Antonakos , J y Mansfield, Kenneth Jr. *Programación Estructurada en C*. España: Prentice Hall.