



SÍLABO

I. INFORMACIÓN GENERAL

Curso	: Fundamentos de programación
Código	: CC112
Prerrequisito	: BIC01
Dpto. Académico	: Ciencia de la Computación
Condición	: Obligatorio
Ciclo Académico	: 2025-1
Créditos	: 4
Horas teóricas	: 2 horas semanales
Horas prácticas	: 4 horas semanales
Sistema de Evaluación	: G
Profesores de Teoría	: E. Cuadros, A. Chulluncuy, Z. Mamani, P. Montalvo.
Profesores de Práctica	: E. Cuadros, A. Chulluncuy, C. Díaz, Z. Mamani, L. Roca.

II. SUMILLA

Este curso está diseñado para dotar a los estudiantes de un conjunto de habilidades en la programación moderna de C++, enfocándose tanto en la comprensión teórica como en la aplicación práctica. Las competencias desarrolladas a través de este curso preparan a los estudiantes para los desafíos complejos del desarrollo de software, enfatizando las prácticas de código limpio, la gestión eficiente de la memoria y las metodologías robustas de depuración y pruebas. Las cuatro unidades de aprendizaje proporcionan un camino estructurado desde técnicas avanzadas de programación hasta la gestión de proyectos y compilación, asegurando que los estudiantes estén bien preparados para roles profesionales en el desarrollo de software en un aspecto inicial.

III. COMPETENCIAS

- El estudiante desarrollará programas que involucren el procesamiento de cadenas de caracteres y la manipulación de archivos, aplicando las funciones pertinentes del lenguaje de programación.
- El estudiante aplicará los principios de la programación orientada a objetos para diseñar y desarrollar sistemas modulares y reutilizables, utilizando clases, objetos, herencia, polimorfismo y encapsulamiento.
- El estudiante comprenderá sobre la gestión de memoria dinámica y las mejores prácticas en la gestión de recursos para prevenir fugas de memoria y asegurar un uso eficiente de la memoria.

IV. UNIDADES DE APRENDIZAJE

CAPÍTULO 1: ITERACIÓN Y RECURSIVIDAD / 6 HORAS

Semana 1:

Prueba de entrada ABET

Recursividad e iteración (24/03/25-29/03/25): Funciones recursivas versus funciones iterativas.

CAPÍTULO 2: ORDENAMIENTO Y BÚSQUEDA / 12 HORAS

Semana 2:

Algoritmos de ordenamiento y búsqueda I (31/03/25-05/04/25) Ordenamiento de arreglos por los algoritmos: burbuja (*Bubble Sort*), selección (*Selection Sort*) e inserción (*Insertion Sort*). Búsqueda lineal.

Semana 3:

Algoritmos de ordenamiento y búsqueda II (07/04/25-12/04/25): Ordenamientos de arreglos por los métodos ordenamiento de mezcla (*Merge Sort*) y ordenamiento rápido (*Quick Sort*). Búsqueda binaria.

Semana 4:

Primera práctica calificada: martes 15/04/25: Temas: Recursión e iteración. Algoritmos de ordenamiento I, II y búsqueda I y II.

CAPÍTULO 3: PUNTEROS Y CADENAS / 24 HORAS

Punteros I (14/04/25-19/04/25): Variables y memoria. Puntero a un arreglo de una dimensión. Aritmética de punteros. Arreglo de una dimensión visto como puntero.

Semana 5:

Punteros II (21/04/25-26/04/25): Paso de punteros como parámetro de una función. Arreglo de punteros. Punteros a punteros. Reglas de precedencia de los modificadores *, () y []. Puntero a void, a const. Aplicaciones de punteros: Paso de argumentos por referencia, navegación de arreglos de una dimensión. Punteros inteligentes.

Semana 6:

Cadena de caracteres I (28/04/25-03/05/25): Caracteres ASCII. Aritmética limitada de caracteres. Arreglo y Cadena de caracteres. Puntero a cadena de caracteres. Funciones de entrada de datos desde el teclado. Funciones de salida de información. Funciones para el manejo de cadenas I.

Semana 7:

Cadena de caracteres II (05/05/25-10/05/25): Funciones para el manejo de cadenas. Funciones para caracteres. Funciones para convertir cadenas de caracteres

Segunda práctica calificada: martes 06/05/25: Temas: Punteros I, II. Cadenas I, II.

Semana 8:

Parcial: 12-17/05/25. Temas: Recursividad e iteración. Ordenamiento y Búsqueda. Punteros y cadenas.

CAPÍTULO 4: MANEJO EFICIENTE DE MEMORIA 18/HORAS

Semana 9:

Memoria dinámica con arreglos (19/05/25-24/05/25): El área de memoria dinámica de la computadora. Asignación de espacio: el operador new. Liberación de espacio: el operador delete. Arreglo de una dimensión utilizando punteros y alojamiento de memoria. Arreglos multidimensionales utilizando punteros y alojamiento dinámico de memoria.

Semana 10:

Estructuras (26/05/25- 31/05/25): Definir, asignar, leer e imprimir variables de tipo struct. Arreglo de estructuras. Anidamientos de estructuras. Puntero a estructura. Paso de estructura como parámetro de una función. Funciones y estructuras.

Semana 11:

Tercera práctica calificada: martes 03/06/25. Temas: Memoria dinámica con arreglos, estructuras, puntero a estructuras. Funciones y estructuras.

Gestión dinámica de memoria y estructuras (02/06/25-07/06/25): Problemas con la asignación de memoria. Resumen del uso de punteros y memoria dinámica.

CAPÍTULO 5: ARCHIVOS / 6 HORAS

Semana 12:

Archivos (09/06/25-14/06/25): Nociones sobre archivos. Archivos físicos y archivos lógicos. Archivos de texto. Organización de archivos y modos de acceso. Operaciones sobre los archivos. Mantenimiento de Archivos. Archivos binarios.

Semana 13:

Cuarta práctica calificada: martes 17/06/25. Temas: Gestión de memoria y estructuras. Archivos.

CAPÍTULO 6: INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS/ 12 HORAS

Clases (16/06/25-21/06/25): Introducción al uso de clases y objetos: atributos y métodos. Constructores y destructores. Uso del puntero this en un constructor.

Semana 14:

Introducción a la Programación Orientada a Objetos (23/06/25-28/06/25) ¿Qué es la Programación Orientada a Objetos? Herencia. Tipos de herencia. Implementación de la herencia. Introducción al Polimorfismo.

Semana 15

Repaso (30/06/25-05/07/25): Clase de repaso o recuperación

Quinta práctica calificada: martes 01/07/25. Clases y programación orientada a objetos.

Semana 16:

Final: 07-12/07/25. Temas: Cadenas de caracteres II. Gestión dinámica de memoria. Archivos. POO.

Semana 17:

Sustitutorio: 14-19/07/25. Temas: Iteración y recursividad. Ordenamiento y búsqueda. Punteros. Cadenas. Gestión dinámica de memoria. Archivos. POO.

V. METODOLOGÍA

Método presencial de aprendizaje: 2h/semana de teoría y 4h/semana de práctica en las cuales el profesor explica e induce las bases teóricas, complementada con aplicaciones preferentemente relacionadas a la especialidad respectiva.

VI. SISTEMA DE EVALUACIÓN

Sistema de evaluación G:

- 1) Son 5 prácticas (P1, P2, P3, P4, P5) y exámenes: parcial (EP), final (EF) y opcionalmente sustitutorio (ES). Se califica sobre 20 puntos.
- 2) PP = Elimina la práctica de menor nota y promedia las otras 4.
- 3) Promedio del curso (PC):

$$PC = [(PP + EP + EF) / 3].$$

$$PC = [(PP + \text{máxima nota (EP, EF)} + ES) / 3]$$

Si no se rinde ES

Si se rinde ES.

VII. BIBLIOGRAFÍA

- Lospinoso, J. (2019). C++ Crash Course A Fast-Paced Introduction. No starch press San Francisco.
- Modern C++: Efficient and Scalable Application Development Richard Grimes, Marius Bancila 2018 Publisher(s): Packt Publishing
- Modern C++ For Software Developers Serious C++ Development Karan Singh, M.Sc., LL.b, Attorney at Law
- C++ High Performance: Master the art of optimizing the functioning of your C++ code, Bjorn Andrist, Viktor Sehr, Ben Garney. Packt Publishing 2020.
- C++ Primer (5th Edition) – Lippman, Lajoie & Moo.
- Effective Modern C++ – Scott Meyers.
- The C++ Programming Language (4th Edition) – Bjarne Stroustrup.
- Deitel, P., & Deitel, H. (2017). C++ How to Program (10th Edition). Pearson