

Facultad de Ingeniería y Ciencias
Escuela de Informática y Telecomunicaciones

PROGRAMA DE ASIGNATURA
Programación Avanzada

I. Identificación

- Nombre : Programación Avanzada
- Código : CIT1010
- Créditos : 6
- Duración : Semestral
- Ubicación en plan de estudio: Semestre 2
- Requisitos : CIT-1000 Programación
- Sesiones semanales : 2 cátedras, 1 ayudantía

II. Descripción del curso

El curso se centra en el manejo de conceptos y elementos avanzados de programación que permitan usar el computador para resolver problemas.

Además, al finalizar el curso el/la estudiante deberá ser capaz de:

1. Utilizar el pensamiento algorítmico para resolver un problema planteado.
2. Aprender los fundamentos y conceptos básicos de la programación orientada a objetos.
3. Desarrollar capacidades de abstracción y modelamiento utilizando estructuras de almacenamiento avanzadas y paradigmas de programación modernos.

III. Resultados de aprendizaje

Al finalizar el curso el/la estudiante será capaz de:

1. Diseñar programas, aplicando los fundamentos y conceptos básicos de la programación orientada a objetos, para resolver problemas del área ingenieril.
2. Desarrollar programas utilizando estructuras de almacenamiento avanzadas y paradigmas de programación modernos, para el correcto manejo, presentación y análisis de información.
3. Diseñar e implementar proyectos relacionados al procesamiento automático de información, individual o grupalmente, documentando el diseño de la solución planteada.

IV. Unidades Temáticas

1. **Revisión de elementos de programación:** Elementos básicos de un lenguaje de programación, estructuras de control, funciones, paso de parámetros, funciones del

lenguaje de programación, arreglos, búsqueda, archivos secuenciales.

2. **Orientación a objetos**

- a. **Introducción a objetos:** Definición y creación de objetos, atributos y ámbitos de estos (públicos y privados). Constructores y métodos.
 - i. Conocer la sintaxis de definición de clases, atributos y métodos.
 - ii. Aplicar el concepto de clases para el modelado y creación de objetos.
 - iii. Comprender los ámbitos de llamado de métodos y atributos según el ámbito de acceso a estos.
 - iv. Transferencia de conocimiento a elaboración de modelado más complejo mediante ejemplos y ejercitación.
- b. **Manejo de instancias:** Creación de múltiples instancias de una clase, interacción entre objetos, objetos como valores de retorno, objetos como parámetros.
 - i. Comprender la diferencia entre distintas instancias de una misma clase.
 - ii. Comprender el mecanismo de acceso a métodos y atributos de objetos y la interacción entre ellos.
 - iii. Comprender el uso y aplicación de objetos como parámetros y valores de retorno.
 - iv. Transferencia de conocimiento a elaboración de modelado más complejo.
- c. **Herencia y polimorfismo:** Concepto de herencia y polimorfismo, casting y sobrecarga de métodos, operadores y templates.
 - i. Comprender el concepto e implicancias de herencia en el modelado de soluciones.
 - ii. Conocer la sintaxis de casting, sobrecarga de métodos, operadores y templates.
 - iii. Transferencia de conocimiento a elaboración de modelado más complejo.

3. **Modelado utilizando contenedores de información:** Manejo de tipos de datos abstractos: vector, lista, cola, cola de prioridad, pila, diccionario.
 - a. Conocer estructuras de almacenamiento y modelado avanzado.
 - b. Modelado y métricas de eficiencia en base a las herramientas utilizadas.
 - c. Comprender las ventajas y desventajas de la utilización de distintas estructuras de almacenamiento en términos de uso de memoria y tiempos de operación.
 - d. Resolver problemas contextualizados simples de la vida cotidiana, utilizando las herramientas del curso.

V. Metodología

Se contempla la realización de 2 sesiones de cátedra en aula de clases. Además, se dispone de tutoría en Laboratorio, donde se aplicará la técnica de estudio de casos, en las que las materias del curso se irán desarrollando en base a los ejemplos aplicados que en cada sesión se planteen.

Durante el semestre se realizarán evaluaciones en laboratorios de las unidades temáticas vistas hasta la fecha de cada evaluación. Además, realizará un trabajo relacionado a la unidad de **Modelado utilizando contenedores de información**, el cual contempla la entrega de un informe asociado a la solución diseñada.

El curso será apoyado con apuntes de clases almacenados en un servidor para acceso vía Internet, en el que se describirán las materias y los ejercicios que deberán realizarse clase a clase.

Dado que esta es una asignatura con una orientación práctica, se ha contemplado que la evaluación del curso considere los desarrollos realizados por el/la estudiante sesión a sesión.

VI. Evaluación

Se contempla la realización de evaluaciones parciales (controles, trabajos, prácticas de laboratorio, entre otros), dos pruebas solemnes de igual ponderación y un examen.

Las notas parciales contemplan un mínimo de 3 laboratorios más el trabajo de archivos descrito en la sección metodología, pudiendo incluirse hasta dos evaluaciones parciales adicionales según el criterio del profesor(a) de la asignatura. En cualquier caso, el promedio de notas parciales corresponde al promedio simple entre las evaluaciones antes descritas.

La nota de presentación a examen se calcula de la siguiente forma:

Nota de Presentación = (25% Solemne 1 + 25% Solemne 2 + 20% Notas Parciales)/0.7

A su vez, la nota final del curso está dada por:

Nota Final= 25% Solemne 1 + 25% Solemne 2 + 20% Notas Parciales + 30% Examen.

Según regla general, para aprobar el curso debe tenerse que Nota Final \geq 4,0 y para presentarse a Examen se debe cumplir con una Nota de Presentación \geq 3,5.

La inasistencia a una prueba solemne implicará reemplazo de su nota con la Nota de Examen. El profesor podrá eximir del examen a el/la estudiante cuyos promedios de notas parciales y de pruebas solemnes sean superiores a 5.0.

VII. Bibliografía Básica

1. The C++ Programming Language, special edition, 2000, Bjarne Stroustrup.
2. C++: Manual de referencias con anotaciones.1994, Margaret Ellis.
3. Programación en C++: Un enfoque práctico. 2006, Luis Aguilar.
4. Programación orientada a objetos, 1998, Luis Aguilar.
5. Apuntes de Cátedra.

PAUTAS ETICAS BASICAS

El aula es un espacio donde los intercambios buscan generar un clima que potencie el aprendizaje, basado en el respeto y el buen trato. Las diferencias, tanto entre estudiantes, como entre estudiante y docentes, deben abordarse desde este marco de respeto.

La universidad cuenta con dos reglamentos importantes de conocer:

- *Reglamento de Convivencia*
- *Normativa de Prevención y Sanción de Acciones de Discriminación, Violencia Sexual y/o de Género.*

Puedes consultar los reglamentos aquí: <https://www.udp.cl/universidad/reglamentos-y-politicas/>

El plagio es el uso de las ideas o trabajo de otra persona sin el adecuado consentimiento. El plagio puede ser intencional o no. El plagio intencional es el claro intento de hacer pasar el trabajo o ideas ajenas como el suyo propio para su beneficio. El plagio no intencional puede ocurrir si Ud. no conoce el mecanismo adecuado de referenciar la fuente de sus ideas e información. Si no está seguro de los métodos aceptados para referenciar, debería consultar con su profesor, tutor o personal de biblioteca.

El plagio comprobado es una actitud que puede resultar en severas sanciones disciplinarias y/o en la exclusión de la Universidad (Artículo 44, Reglamento del Estudiante de Pregrado).

Elaborado por: Jonathan Frez

Revisado por:

Fecha revisión: Enero 2022

Fecha vigencia: Marzo de 2022