

SYLLABUS de la Asignatura

Programación 2

1. Descripción de la asignatura

Esta asignatura aporta al perfil del Ingeniero en Computación e Informática la capacidad de analizar, desarrollar, implementar y administrar software de aplicación orientado a objetos con un nivel de complejidad mayor que el curso que le precede, cumpliendo con estándares de calidad, con el fin de apoyar la productividad y competitividad de las organizaciones.

Además proporciona soporte a otros ramos, más directamente vinculadas con desempeños profesionales; se ubica en el segundo semestre de la malla curricular. Proporciona al estudiante las competencias necesarias para abordar el estudio de cualquier lenguaje orientado a objetos, metodología de análisis y diseño orientado a objetos, manejo de archivos y colecciones.

- **Sentido y ubicación de la asignatura en el plan de estudios**

Programación II tiene como propósito profundizar los conocimientos de la programación orientada a objeto y la forma de cómo manejar la información de manera temporal a través de colecciones, encontrándose en el segundo nivel del plan de estudios

2. Prerrequisitos, Co-requisitos y Horas Pedagógicas

Curso	:Programación I
Código	: INS124
Tipo de actividad	: Teórico y Práctica
Horas semanales	: 6
Requisitos	: Programación I
Co-requisitos	:Lenguaje de Programación

3. Aprendizajes Esperados

El estudiante desarrollará aprendizajes que le permitirán:

- Diseñar e implementar objetos de programación que permitan resolver situaciones reales y de ingeniería.
- Desarrollar aplicaciones a través del manejo de eventos
- Aprender a usar colecciones o mecanismos de almacenamiento tanto temporal como arreglos unidimensionales y bidimensionales.

- Aprender a manejar errores en tiempos de ejecución a través del uso de excepciones.
- Implementar los conceptos asociados a la programación orientada a objeto como clases, métodos, interfaces, herencia, polimorfismo, sobrecarga, etc.
- Desarrollar aplicaciones que hagan uso de Bases de Datos. (Proyecto Semestral)

4. Sistema de Evaluación de la Asignatura

4.1 Para la evaluación de las competencias cognitivas se sugiere:

- Preguntas de desarrollo.
- Talleres grupales
- Trabajos de investigación
- Desarrollo de mapas conceptuales

4.2 Para la evaluación de las competencias procedimentales se sugiere:

- Observación sistemática.
- Escalas de valoración o apreciación
- Pruebas de realización de tareas prácticas
- Diario de clases
- Resolución de casos
- Presentaciones orales
- Confección de informes

4.3 Para la evaluación de las competencias actitudinales se sugiere:

- Entrevistas
- Cuestionario de opinión
- Trabajo en equipo
- Participación en clases

Nota de Presentación: a examen se calcula como sigue:

Ítem	Porcentaje	Actividad	Responsable
Solemne N°1	20%	Individual	Profesor titular
Solemne N°2	30%	Individual	Profesor titular
Ayudantía	25%	Individual	Profesor ayudante
Proyecto	25%	Grupal	Profesor titular
Control de Salida		El alumno	

		<p>rinde</p> <p>Una evaluación individual en la cual la nota máxima corresponde a la nota obtenida en el trabajo del proyecto.</p> <p>Esta podría ser un control escrito.</p>	
--	--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Nota de Laboratorio: Se calcula como sigue:

Ítem	Porcentaje	Actividad	Responsable
Taller 1		Individual	Profesor ayudante
Taller 2		Individual	Profesor ayudante
Taller 3		Individual	Profesor ayudante
Talleres	40%	Se elimina un control	
Proyecto	60%	Grupal	Profesor ayudante
		<p>Control salida</p> <p>El alumno rinde</p> <p>Una evaluación individual en la cual la nota máxima corresponde a la nota obtenida en el trabajo del proyecto.</p> <p>Esta podría ser un control</p>	

		escrito.	
--	--	----------	--

Condiciones transversales:

Ayudantía:

- De la totalidad de los talleres se elimina la nota más baja.

Proyectos:

Si el alumno no se presenta a los controles de salida de los proyectos, su participación se evalúa con nota mínima.

Prueba Recuperativa Justificada: Si al alumno ha justificado una solemne, la nota 1.0 será reemplazada por la nota de examen.

Eximición: Si la nota de presentación a examen es superior o igual a 5,0 puntos, el alumno o alumna podrá eximirse de la rendición del examen de la asignatura.

Si una nota < 4.0, se exige con 5.5.

Los alumnos que no se eximan calcularán su calificación final de la siguiente manera:

Nota Final: Nota de Presentación * 70% + Nota de Examen * 30%

5. Actividades

Tipo de Actividad	Sistema de Retroalimentación	Formato y medio de entrega de tareas	Política de entregas atrasadas
Trabajos individuales	Todos los trabajos serán evaluados durante el desarrollo de la asignatura.	Los estudiantes deberán subir sus trabajos finales en el aula virtual y/o en plataforma git.	No se aceptarán entregas atrasadas.
Talleres	Todos los talleres se realizarán individual, donde cada uno de ellos contendrá una o más unidades.	Los estudiantes deberán entregar el desarrollo de cada taller en horario de clases en el aula virtual y/o en plataforma git.	Todo alumno que falte a un taller no tendrá derecho a recuperarlo.
Proyecto semestral (parejas)	Será evaluado al final del semestre	Los estudiantes deberán subir sus	No se aceptarán entregas atrasadas y

		trabajos finales en el aula virtual y/o en plataforma git.	trabajos iguales
--	--	------------------------------------------------------------	------------------

6. Contenidos

1. Unidad: Creación Clases y Métodos

- Describir la ventaja de las clases
- Describir la ventaja de los métodos.
- Declarar clases y métodos. Invocar métodos
- Comparar objetos y métodos estáticos.
- Usar métodos con sobrecarga
- Recursividad

2. Unidad: Herencia, encapsulamiento y excepciones

- Definir y comprobar el uso de herencias, clase, superclase
- Explicar el concepto de abstracción
- Explicar e identificar las bibliotecas usadas en el programa
- Explique y use el concepto de interface
- Explique y use el concepto de polimorfismo
- Excepciones
- Colecciones

3 Unidad: Arreglos unidimensionales y bidimensionales

- Arreglos unidimensionales
- Declaración, instanciación, inicialización
- Uso de atributos de los arreglos.
- Uso en bucles de arreglos
- Crear arreglos bidimensionales.
- Declaración, instanciación, inicialización

4.- Unidad: Manejo de Archivos y Base de Datos.

- Utilizar archivos para el almacenamiento y lectura de información.
- Manejo de Base de Datos:
 - Conexión y uso de Base de datos:

Sedes: Santiago, Viña y Concepción
www.unab.cl

- Crear Base de Datos,
- Conexión a Base de Datos,
- Insertar, Consultar, modificar, eliminar registros
- Uso de vistas y/o procedimientos almacenados para obtener el proceso datos requerido.

5. Unidad: Programación a través de eventos y proyecto semestral

- Introducción a los proyectos visuales
- Objetos comunes y sus atributos (botones, cuadros de texto, listas de selección)

7. Calendario de Actividades, uso de lenguaje Java.

Semana	Actividad	Contenido
1	<p>Presentación de Objetivos Ramos e introducir conceptos básicos.</p> <p>Prueba de diagnóstico</p> <p>Ejercicios de repaso y desarrollo de la prueba de diagnóstico</p>	Repaso
2	<p>Programación a través de clases y métodos, instancia de métodos.</p> <p>Métodos con o sin retorno</p> <p>Ejercicios de programación</p>	<p>Explicar el concepto de abstracción</p> <p>Describir la ventaja de las clases y métodos.</p> <p>Declarar e invocar métodos</p>
3	<p>Ejercicios de programación que hagan uso del concepto de sobrecarga o polimorfismo y métodos estáticos, variables estáticas.</p>	<p>Comparar objetos y métodos estáticos.</p> <p>Usar métodos con sobrecarga</p>
4	<p>Taller 1 (ayudantía)</p> <p>Evaluación del taller</p> <p>Implementar el concepto de herencia</p>	<p>Definir y comprobar el uso de herencias, clase, superclase</p>
5	<p>Ejercicios de programación que hagan uso de interface y excepciones</p>	<p>Explicar e identificar las bibliotecas usadas en el programa</p> <p>Explicar el concepto de</p>

		interface y excepciones Excepciones
6	Ejercicios de programación que hagan uso de interface y excepciones <u>Prueba Solemne 1</u>	Explicar e identificar las bibliotecas usadas en el programa Explicar el concepto de interface Excepciones
7	Revisar Solemne 1 Taller 2 (ayudantía) Ejemplos e implementación del uso de colecciones	Colecciones, clases que hacen uso de Collection Ventaja y desventaja de las colecciones
8	Ejemplos e implementación del uso de colecciones	
9	Ejemplos e implementación de arreglos de una y dos dimensiones	Arreglos unidimensionales Declaración, instanciación, inicialización
10	Ejercicios de programación, uso de arreglos, crear clases de arreglo, métodos Implementación de matrices Ejercicios de matrices Introducción a aplicaciones visuales en Java con SWING Introducción a los proyectos visuales en Java son SWING Objetos comunes y sus atributos (botones, cuadros de texto, listas de selección) Taller 3 (ayudantía)	Uso de atributos de los arreglos. Uso de arreglos de una y dos dimensiones Programación Visual basada en eventos
11	<u>Entrega Rubrica de Proyecto por evaluar en Catedra</u> <u>Prueba solemne 2</u>	Programación Visual, basada en eventos
12	Objetos comunes y sus atributos (botones, cuadros de texto, listas de selección) Utilización de interfaz gráfica para almacenar	Programación Visual, basada en eventos Programación visual con formas de almacenamiento

	datos en estructuras de datos simples como vectores o matrices Manejo de Archivos con estructura lógica: Creación, escritura, lectura.	
13	Desarrollo Aplicaciones con acceso a Base de Datos en ambiente consola y /o Grafico Desarrollo Aplicaciones con acceso a Base de Datos en ambiente Grafico	
14	Desarrollo Aplicaciones con acceso a Base de Datos en ambiente Grafico	
15	Revisión de Proyectos Semestrales	
16	Examen	
17	Revisión de Examen	

8. Bibliografía

Unidad	Obligatorio	Complementario	Código Biblioteca/Ubicación
1	Joyanes Aguilar, L. (2010). Programación en C, C++, Java y UML. Mexico: McGraw-Hill Interamericana.		
2	Joyanes Aguilar, L. (2003). Fundamentos de programación: Algoritmos, estructura de datos y objetos. Joyanes Aguilar, L. (2010). Programación en C, C++, Java y UML. Mexico: McGraw-Hill Interamericana..		
3	Joyanes Aguilar, L. (2003). Fundamentos de programación: Algoritmos, estructura de datos y objetos. Joyanes Aguilar, L. (2010). Programación en C, C++, Java y UML. Mexico: McGraw-Hill Interamericana.	.	
4	Apunte de la clase en aula virtual/intranet		