Programa de Asignatura

1. Identificación Asignatura

<table>
<tr><td>Nombre:</td><td colspan="3">Programación Avanzada
Programación II</td><td>Código:</td><td>IF 1002
IF 1015</td></tr>
<tr><td>Carrera:</td><td colspan="2">Ingeniería Civil Informática
Ingeniería Civil Industrial</td><td>Unidad Académica:</td><td colspan="2">Ciencias Naturales y Tecnología</td></tr>
<tr><td>Ciclo Formativo:</td><td colspan="2">Ciclo Inicial</td><td>Línea formativa:</td><td colspan="2">Básica</td></tr>
<tr><td>Semestre</td><td colspan="2">III</td><td>Tipo de actividad:</td><td colspan="2">Obligatoria</td></tr>
<tr><td rowspan="2">N° SCT:</td><td rowspan="2">6</td><td colspan="4">Horas Cronológicas Semanales</td></tr>
<tr><td>Presenciales:</td><td>4.5</td><td>Trabajo Autónomo:</td><td>4.5</td></tr>
<tr><td>Pre-requisitos</td><td colspan="5">Fundamentos de Programación</td></tr>
</table>

2. Propósito formativo

La asignatura de Programación Avanzada tiene como propósito que el/la estudiante aprenda a desarrollar soluciones de programación siguiendo el paradigma orientado a objetos (POO).

En su primera parte, la asignatura aborda conceptos de orientación a objetos al mismo tiempo que los aplica usando el lenguaje de programación Python. Luego, en una transición al lenguaje C++ el/la estudiante ira conociendo y aplicando elementos como objetos, clases, herencia, abstracción y polimorfismo, entre otros. El contexto del desarrollo de habilidades de aplicación metodológica será a través de problemas específicos, definidos en diversos dominios de aplicación cuyas soluciones se encontrarán delimitadas en cuanto al alcance y complejidad.

Esta asignatura plantea nuevos paradigmas de programación a los vistos hasta este momento, y promueve su aplicación en una asignatura práctica del mismo nivel que es Taller de Ingeniería III. De forma incipiente, los y las estudiantes comenzarán a trabajar con diferentes estructuras de datos que serán profundizadas en la siguiente asignatura de especialidad que es Algoritmos y Estructuras de Datos.

3. Contribución al perfil de egreso

Esta asignatura contribuye a los siguientes desempeños o resultados de aprendizaje globales declarados en el Perfil de Egreso de la carrera:

1. Entiende problemas a través de la construcción de abstracciones conceptuales, cualitativas y cuantitativas, utilizando formalismos establecidos, que permitan formular soluciones
2. Diseña y programa soluciones, utilizando estrategias algorítmicas, que permitan resolver problemas de forma eficaz y acorde a múltiples objetivos de diseño.

4. Resultados de aprendizaje específicos

Resultado de Aprendizaje Específico	Criterios de evaluación	Evidencia

RA1. Reconoce los componentes del paradigma de programación orientado a objetos, definiendo cada una de ellos y reconociéndolas en contextos de uso.	1.1. Comprende los conceptos del paradigma de programación orientada a objetos tales como: polimorfismo, encapsulamiento, herencia, sobrecarga, funciones virtuales, etc. 1.2. Define y diagrama clases y sus componentes: campos, constructor, métodos accesores y mutadores. 1.3. Modelar problemas utilizando el paradigma de programación orientada a objetos.	Laboratorios, guias de ejercicio.
RA2. Aprende a utilizar el lenguaje de programación C++ a través de la aplicación de los componentes que subyacen el paradigma de programación orientado a objetos.	1.4. Identifica diferencias entre los lenguajes de programación Python y C++. 1.5. El/la estudiante logra elaborar de manera autónoma programas en C++ utilizando el POO que resuelven problemas simples.	Laboratorios, guias de ejercicio.
RA3. Programa/utiliza funciones/librerías en C++ considerando clases y objetos, para la la creación de soluciones de software de propósito general, adaptando código existente para satisfacer requerimientos nuevos o desarrollando aplicaciones a partir de requerimientos nuevos.	1.6. El/la estudiante logran desarrollar soluciones de forma autónoma a problemas de diversa complejidad utilizando el lenguaje C++ y con un POO.	Laboratorios, guias de ejercicio.

5. Unidades de Aprendizaje

1. Introducción a la modelación orientada a objetos – relaciones de asociación 1.1. Paradigmas de programación 1.2. Lenguajes orientados a objetos 1.3. Identificación y evolución de los esquemas de abstracción. 1.4. Conceptos de clase, atributo, operación y objeto. 1.5. Aliasing, manejo de referencias
2. Programación Orientada a Objetos en C++ 2.1. Introducción al lenguaje C++ 2.2. Aplicación de Abstracción: Clases, Atributos, Operaciones (métodos, clases, herencia) 2.3. Modularidad (criterios, principios y reglas) 2.4. El concepto de interfaz 2.5. El concepto de objeto 2.6. Asociaciones, agregaciones y composiciones
3. Herencia, Polimorfismo y Gestión de Errores 3.1. Abstracción 3.2. Tipos de jerarquías y costes de la herencia 3.3. Tipos de polimorfismos

3.4. Relaciones entre polimorfismos
3.5. Gestión de errores. Persistencia y concurrencia.

6. Recursos de Aprendizaje

Bibliografía:

B1. Ceballos Sierra, F. J. (2018). Programación orientada a objetos con C++ (5a. ed.). RA-MA Editorial. https://elibro.net/es/lc/uaysen/titulos/106519

B2. Oviedo Regino, E. (2015). Lógica de programación orientada a objetos. Ecoe Ediciones. https://elibro.net/es/lc/uaysen/titulos/70431

Recursos materiales e infraestructura:

- Laboratorio de computación.
- Acceso a Ucampus.
- Acceso a Googlesites con credenciales institucionales.

Computadores debidamente equipados para utilizar lenguajes de alto nivel (por ej.: Python).

7. Comportamiento y ética académica:

Se espera que los estudiantes actúen en sus diversas actividades académicas y estudiantiles en concordancia con los principios de comportamiento ético y honestidad académica propios de todo espacio universitario y que están estipulados en el *Reglamento de Estudiantes de la Universidad de Aysén*, especialmente aquéllos dispuestos en los artículos 23°, 24° y 26°.
Todo acto contrario a la honestidad académica realizado durante el desarrollo, presentación o entrega de una actividad académica del curso sujeta a evaluación, será sancionado con la suspensión inmediata de la actividad y con la aplicación de la nota mínima (1.0).

Planificación del curso

8. Responsables

Académico (s) Responsable (s) y equipo docente	**Profesor**: **Dra. Mariela I. González Flores**		
Contacto	**Correo**: mariela.gonzalez@uaysen.cl		
Año	2024	**Periodo Académico**	Primer semestre
Horario clases	**Cátedra:** **Lunes: 14:30 a 17:45 hs.** **Viernes: 12:00 a 13.30 hs.**	**Horario de atención estudiantes**	Miércoles 14.30 a 15.30 hs.
Sala / Campus	Sala Virtual Ucampus		

9. Metodología de Trabajo:

La asignatura contiene:			
Actividades de vinculación con el medio	**No**	Actividades relacionadas con proyectos de investigación	**No**

En el curso se contemplan cuatro tipos de actividades docentes, las cuales se asocian a requerimientos de sala y al nivel de intervención del profesor:

Actividad docente	Descripción	Intervención del profesor/ayudante	Requerimiento de sala
Exposición conceptual	El profesor introduce conceptos de programación preliminares y necesarios a otras actividades de índole práctica, de forma expositiva. Se dispone de materiales complementarios en la plataforma Ucampus.	Alta	Sala de clases UCAmpus Online UCampus Offline
Programación expositiva	El profesor profundiza en la comprensión de elementos conceptuales a través de la exposición directa de la resolución de problemas de programación como ejemplos.	Alta	Sala de clases UCAmpus Online UCampus Offline
Programación tutorial	Funciona como la programación expositiva, pero el profesor realiza pausas para que los alumnos completen "pasos requeridos" antes de continuar. El objetivo es que todos los alumnos completen un paso definido por el profesor antes de continuar al siguiente.	Media	Laboratorio de computación Computador persona
Actividad práctica / Programación autónoma	Los estudiantes abordan y resuelven problemas de programación de forma autónoma, algunas con guía y apoyo docento y otras no.	Baja/Media	Laboratorio de computación Computador persona

En cualquier semana del semestre en curso se podría realizar una **evaluación menor** sobre las temáticas estudiadas a la fecha. Esta evaluación menor puede ser de los siguientes tipos:

- **Laboratorio**: Evaluación individual o grupal, que se realiza en el computador. Ocupará los bloques del día jueves.
- **Guía de ejercicios**: Evaluación individual que se realiza en computador durante el tiempo de trabajo autónomo.
- ***Prueba Parcial:*** Evaluación individual que se realiza en computador en el horario de clases.
- ***Proyecto:*** Evaluación individual o grupal, que se realiza en el computador. Ocupará los bloques horario autónomo.

10. Evaluaciones:

Evaluación	Ponderaciones específicas	Ponderación nota presentación
Pruebas de cátedra	• Prueba Unidad 1 (P1): 30% , **8 abril del 2024** • Prueba Unidad 2 (P2): 35%, **13 de mayo del 2024** • Prueba Unidad 3 (P2): 35%, **1 de julio del 2024**	70%

Laboratorios y Guías	El promedio simple entre las notas consideradas corresponde al 100%.	30%

Calificación final:
- Nota de presentación: 70%
- Examen Final: 30 %

Condiciones de eximición:
- Nota de presentación igual o superior a nota 5,0
- Ponderación específica pruebas de cátedra >= 4,0

Derecho a rendir examen:
Nota de presentación >= 3,5

Requisito de Aprobación
- Asistencia: 70%
- Nota Final: 4,0

11. Otros aspectos asociados al funcionamiento del curso:

- Toda la coordinación del curso (comunicaciones, actualización de notas, material, etc.) será realizada a través de UCampus. El estudiante deberá informar con tiempo suficiente si presenta dificultades de conexión para trasladar el requerimiento a la coordinación de programa. Adicionalmente los estudiantes deberán acceder al material disponible en el google sites que el profesor preparó para este propósito.

Es deber del estudiante mantenerse informado de las noticias, avisos y material entregado por el profesor a través de estos medios, se sugiere instalar en su dispositivo móvil la aplicación de UCampus.

12. Planificación de las actividades de enseñanza- aprendizaje y de evaluación

Semana / Sesión 2024	Resultado(s) de Aprendizaje	Tema (Unidades de aprendizaje) y actividades	Recursos utilizados o lecturas	Actividad(es) de Trabajo Autónomo
Semana 1 (11 al 15 de Marzo)	**RA1.** Reconoce los componentes del paradigma de programación orientado a objetos, definiendo cada una de ellos y reconociéndolas en contextos de uso.	Revisión del programa Acceso al servidor Guía de Linux Unidad 1: 1.1 Paradigmas de programación 1.2 Lenguajes orientados a objetos	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio de Linux

Semana 2 (18 l 22 de marzo)	RA1.	1.3 Identificación y evolución de los esquemas de abstracción. 1.4 Conceptos de clase, atributo, operación y objeto.	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 3 (25 al 28 de marzo)	RA1.	1.5 Manejo de referencias **Unidad 2. Programación Orientada a Objetos en C++**	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 4 (1 al 5 de abril)	RA1.	2.1. Introducción al lenguaje C++	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 5 (8 al 12 de abril)	RA1.	**Prueba 1: 8 abril/2024**		
Semana 6 (15 al 19 de abril)	RA2.	2.2. Aplicación de Abstracción Clases, Atributos.	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 7 (22 al 26 de abril)	RA2.	2.2. Aplicación de Abstracción: Operaciones métodos, clases, herencia.	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 8 (29 de abril al 3 de mayo)	RA2.	2.3 Modularidad (criterios, principios y reglas) 2.4 El concepto de interfaz	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 9 (6 al 10 de mayo)	RA2.	2.5 Concepto de objetos 2.6 Asociaciones, agregaciones y composiciones	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 10 (13 al 17 de mayo)		**Prueba 2: 13 de mayo/2024**		

Semana receso (20 al 24 de mayo)	Receso de clases para los estudiantes			
Semana 11 (27 al 31 de mayo)	RA1.	**Unidad 3:** Herencia, Polimorfismo y Gestión de Errores 3.1 Abstracción 3.2 Tipos de jerarquías y costes de la herencia	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 12 (3 al 7 de junio)		3.3 Tipos de polimorfismos 3.4. Relaciones entre polimorfismos	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 13 (10 al 14 de junio)		3.5. Gestión de errores. Persistencia y concurrencia 3.6 Aplicaciones básicas de Programación orientada a objetos en C++	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 14 (17 al 21 de junio)		3.7 Aplicaciones avanzadas de Programación orientada a objetos en C++	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 15 (24 al 28 de junio)		3.8 Aplicaciones en uso de servidores	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 16 (1 al 5 de julio)		**Prueba 3: 1 de julio 2024**	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 17 (8 al 12 de julio)		**Examen: 12 de julio 2024** **Unidades 1, 2 y 3**	Apunte de clases Libros Acceso a un servidor	
Semana 18 (15 al 19 de julio)		**Revisión de examen y cierre de actas**		