

ACTIVIDAD CURRICULAR DE FORMACIÓN

Facultad o Instituto	:	Ciencias de la Ingeniería
Carrera	:	Ingeniería Civil Informática

I. IDENTIFICACIÓN DE LA ACTIVIDAD CURRICULAR

Nombre	:	Programación Avanzada					
Código	:	INF-223					
Semestre lectivo	:	IV Semestre					
Horas	:	Presencial:	72	Autónomas:	78	TOTAL:	150
Créditos SCT	:	5					
Duración	:	Trimestral:		Semestral:	x	Anual:	
Modalidad	:	Presencial:	x	Semi-presencial:		A Distancia:	
Área de Formación	:	Disciplinar:	x	General:		Profesional:	
						Práctica:	
Pre-requisito (Si los hubiese)	:	Estructuras de datos					

II. DESCRIPCIÓN Y CARACTERIZACIÓN DE LA ACTIVIDAD CURRICULAR

La actividad curricular de Programación Avanzada, corresponde al cuarto semestre del plan de estudios de la carrera de Ingeniería Civil Informática. Ésta pertenece al área disciplinar y es la cuarta asignatura en la línea de programación.

Permite al estudiante comprender y aplicar los principales conceptos y técnicas de la programación orientada a objetos. Un objeto es un tipo de datos con estructura y estado interno. Cada objeto tiene definidas unas operaciones para acceder a ese estado o manipularlo. Este curso es esencialmente una introducción a las técnicas usuales de la Programación Orientada a Objetos. Incluye una parte importante de programación en la cual se implementan en algún lenguaje de programación. Los principales temas tratados son Clases y Objetos, Herencia, Polimorfismo, Concurrencia, Tratamiento de Errores e Interfaces de Usuario.

La metodología utilizada será con clases expositivas-participativas, laboratorios con apoyo de software, aprendizaje colaborativo, aprendizaje basado en problemas, y aprendizaje en base a resolución de problemas.

La evaluación será a través de pruebas escritas, talleres y trabajos de laboratorio y problemas.

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS A LA ACTIVIDAD CURRICULAR.

III.1 COMPETENCIAS PROFESIONALES.

COMPETENCIA	SUBCOMPETENCIA
Aplicar conocimientos de ciencias de ingeniería y ciencia de la computación en el ámbito profesional, utilizando pensamiento crítico y capacidad analítica.	Diseñar soluciones a problemas usando algoritmos, modelos computacionales y ciencias de la ingeniería.
Gestionar proyectos de software, aplicando estándares y metodologías en el desarrollo, operación y mantenimiento de software de calidad que satisfacen necesidades organizacionales	Aplicar estándares y metodologías en el análisis y diseño de Software de calidad en contextos simulados.

III.2 COMPETENCIAS GENÉRICAS.

COMPETENCIA	SUBCOMPETENCIA
Demostrar coherencia ética entre sus postulados valóricos y sus acciones, respetando los derechos humanos y participando activamente en las organizaciones comunitarias, haciendo primar la responsabilidad social desde una perspectiva cristiana.	Actuar comprometido con los derechos humanos, y participa con responsabilidad ciudadana en los distintos escenarios, formales e informales, de la comunidad.
Comunicar ideas, tanto en la lengua materna como en el idioma inglés, haciendo uso de las tecnologías de la información para desenvolverse en diversos escenarios, dando soluciones a diversas problemáticas de la especialidad.	Comunicarse de forma escrita en la lengua materna e inglés de acuerdo a lenguaje académico-profesional haciendo uso de las tecnologías de la información en contextos propios de su profesión.

IV. RESULTADOS DE APRENDIZAJE - APRENDIZAJE ESPERADO.

RESULTADOS DE APRENDIZAJES
1.- Describir los paradigmas de programación vigentes y como estos orientan la resolución de un problema.
2.- Analizar las características de la programación orientada a objetos determinando como estas mejoran la calidad del proceso de programación.
3.- Resolver problemas identificando los objetos principales, describiendo sus atributos y comportamiento.

V. UNIDADES DE APRENDIZAJE Y EJES TEMÁTICOS

R. AP.	UNIDAD	EJE(S) TEMÁTICO(S)
1	Paradigmas de Programación.	Paradigma de Programación estructurada. Paradigma Funcional. Programación orientada a eventos. Programación orientada a objetos. Evolución de los lenguajes imperativos. Identificación de lenguajes de programación y el paradigma sustentado
2	Elementos de la Orientación a Objetos.	Concepto de Clase Concepto de Objeto Concepto de Método Descripción de los conceptos de Clase, objeto y método en una situación de la vida real. Propiedades de abstracción, encapsulación, modularidad, polimorfismo, herencia y jerarquía. Representación gráfica de clases. Diagramas UML
3	Lenguaje orientado a objetos	Creación y diseño de objetos. Utilización de operadores, flujo de control de un programa y palabras reservadas Constructores y destructores Acceso a Clases. Herencia y el polimorfismo. Manejo de Errores. Entrada/Salida Interfaz gráfica.

VI. ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE

De acuerdo al modelo educativo de la Universidad Católica del Maule, la metodología de trabajo para el desarrollo de la actividad curricular, se basa en un enfoque activo-participativo; esto implica entregar un rol protagónico al estudiante que es entendido como eje y centro de acción y quién a través de su participación activa y con orientaciones y lineamientos que le entrega el docente va construyendo su propio aprendizaje. Para lograr este objetivo, las distintas clases consideran una serie de estrategias metodológicas, previamente seleccionadas por el docente, tales como:

- Aprendizaje Colaborativo
- Aprendizaje Basado en problemas
- Aprendizaje en base a resolución de ejercicios y problemas el aula
- Aprendizaje en base a resolución de ejercicios y problemas con uso software en laboratorio computación.
- Método expositivo- participativo

VII. PROCEDIMIENTOS DE EVALUACION DE APRENDIZAJES.

RESULTADO DE APRENDIZAJES	INDICADORES	INSTRUMENTO Y/O TÉCNICA EVALUATIVA	PONDERACIÓN (%)
1	<p>Caracteriza cada uno de los paradigmas de programación.</p> <p>Sitúa cada paradigma en un contexto de evolución de la computación.</p> <p>Desarrolla un mapa conceptual relacionando los distintos paradigmas señalando sus ventajas y desventajas.</p>	<p>Prueba escrita /Pauta</p> <p>Problemas/Rúbrica</p>	20%
2	<p>Identifica en diversas fuentes de información los conceptos principales del paradigma de programación orientado a objetos.</p> <p>Identifica ejemplos de la vida real que apliquen o manifiesten dichos conceptos.</p> <p>Redacta una definición propia de los conceptos de forma simple y entendible.</p> <p>Comenta en clase las definiciones de otros compañeros para enriquecer la propia y consensar una grupal.</p> <p>Analiza analogías taxonómicas de los seres vivos que compartan rasgos comunes por estar relacionados mediante una herencia genética e identificar la especie a la que pertenecen.</p>	<p>Prueba escrita /Pauta</p> <p>Problemas Laboratorio/Rúbrica</p>	20%

	<p>Identifica los atributos y comportamientos propios de una especie que comparten los animales pertenecientes a ella.</p> <p>Analiza objetos reales que compartan características comunes por pertenecer a una misma categoría de objetos.</p> <p>Identifica los atributos y comportamientos propios de una categoría de objetos que compartan todos sus miembros.</p> <p>Analiza la información del lenguaje UML referente al modelado de clases.</p> <p>Diseña diagramas de clases aplicados a distintos problemas.</p>		
3	<p>Programa clases con atributos públicos para exponer y comprender la vulnerabilidad de los datos.</p> <p>Utiliza modificadores de acceso privado o protegido para proteger los atributos.</p> <p>Programa métodos públicos para otorgar acceso seguro a los atributos.</p> <p>Utiliza encapsulamiento para resolver un problema en particular.</p> <p>Programa constructores y destructores para las clases, de manera que permitan dar un valor inicial a sus atributos cuando nazcan sus</p>	<p>Prueba escrita /Pauta</p> <p>Informe Laboratorio/Rúbrica</p>	30%

	<p>objetos, o liberar recursos cuando mueran los mismos.</p> <p>Identifica los comportamientos de una clase que pueden variar dependiendo del paso, cantidad, tipo u orden de argumentos.</p> <p>Programan cada variación del comportamiento en métodos sobrecargados para agregar flexibilidad a la clase.</p> <p>Selecciona un operador existente del lenguaje y sobrecargarlo en la clase de los objetos para implementarles dicha funcionalidad.</p> <p>Crea varias instancias de clases derivadas diferentes para verificar la existencia de los miembros heredados comunes en todas ellas, y la diversidad de sus especializaciones.</p> <p>Redefine un método en una clase derivada para sobrescribir el de su clase base e introducirse al polimorfismo.</p> <p>Implementa clases abstractas en clases base que no requieran ser instanciadas con al menos un método abstracto para que sea implementado por sus clases derivadas en múltiples formas.</p> <p>Crea un programa que deliberadamente genere excepciones comunes para identificar: sus nombres, sus</p>		
--	--	--	--

	<p>causas, su comportamiento, y reporte de error.</p> <p>Programa el lanzamiento, propagación y manejo de una excepción definida por el usuario.</p> <p>Programa una clase que cree, consulte, modifique y borre archivos de texto.</p> <p>Programan una clase que cree, consulte, modifique y borre archivos binarios.</p> <p>Implementa una aplicación con interfaz gráfica orientada a objetos.</p>		
	Todos los aspectos vistos en el semestre	Prueba Acumulativa Final/Pauta	30% del total de la evaluación

VIII. RECURSOS DE INFRAESTRUCTURA

Sala de clases, laboratorio de computación, computadores, proyectores, telones, pizarras amplias, compiladores e intérpretes, Biblioteca, Sistema LMS-UCM.

IX. RECURSOS BIBLIOGRÁFICOS

	Autor, Título, Editorial, Año de Edición	Biblioteca donde se encuentra	N° Libros Disponibles
BÁSICA OBLIGATORIA	- Debrauwer, Laurent, "UML 2, Iniciación, ejemplos y ejercicios corregidos", Ediciones ENI, Barcelona, España, 2013.	-Talca	-5
	-Timothy A. Budd; An Introduction to Object-Oriented Programming; Addison-Wesley, (Tercera Edición), Estados Unidos, 2002.	-Talca	-0
	- Deitel, Cómo Programar en Java, México: Prentice Hall, 1998.	-Talca	-8
	- Joyanes Aguilar, Luis; Rodríguez Baena, Luis; Fernández, Matilde: Fundamentos de Programación,	-Talca	-3

	Algoritmos, Estructuras de Datos y Objetos. McGraw-Hill, 2003.		
	- Paul Deitel, Harvey Deitel, Java : How to program, 10 Edition, New. Jersey : Pearson, 2012.	-Talca	-2
	- Joyanes Aguilar, Luis, Programación en JAVA 6: algoritmos, programación orientada a objetos e interfaz gráfica de usuario, México: McGraw-Hill, 2011.	-Talca	-2
	- Joyanes Aguilar, Luis, Estructuras de datos en Java, Madrid: McGraw - Hill Interamericana, 2008.	-Talca	3

X. OTROS RECURSOS

Nombre Recurso	Tipo de Recurso
Bahit, Eugenia (2012), "Python para Principiantes", http://www.cursosdeprogramacionadistancia.com/static/pdf/material-sin-personalizar-python.pdf (Consulta Julio 2015). Licencia CC.	<i>Sitio WEB</i>