



DATOS DE IDENTIFICACIÓN DEL CURSO

DEPARTAMENTO:	Departamento de Ciencias Computacionales				
ACADEMIA A LA QUE PERTENECE:	Técnicas Modernas de Programación				
NOMBRE DE LA MATERIA:	Taller de Programación Orientada a Objetos				
CLAVE DE LA MATERIA:	CC201				
CARÁCTER DEL CURSO:	Básica Común				
TIPO DE CURSO:	Taller				
No. DE CRÉDITOS:	4				
No. DE HORAS TOTALES:	60	Presencial	48	No presencial	12
ANTECEDENTES:	Se recomienda : CC102 – Introducción a la Programación				
CONSECUENTES:	Se recomienda para CC208 – Lenguajes de Programación Comparados				
CARRERAS EN QUE SE IMPARTE:	Licenciatura en Informática Licenciatura en Ingeniería en Computación				
FECHA DE ULTIMA REVISIÓN:	Agosto 2009				

PROPÓSITO GENERAL

El alumno desarrollará habilidades de diseño y codificación orientada a objetos, utilizando herramientas de programación y modelado actuales, en base a las practicas que se desarrollaran en el curso, se veran las diferentes formas y metodos de soluciones para los problemas presentados en base a los distintos modelos.

OBJETIVO TERMINAL

El alumno utilizará el lenguaje unificado de modelado (UML) para comprender planos de software y lenguaje de programación orientada a objetos para llevar a cabo sus implementaciones.

CONOCIMIENTOS PREVIOS

Programación estructurada.

HABILIDADES Y DESTREZAS A DESARROLLAR

Resolver, abstraer, aplicar, desarrollar, analizar, diseñar, modelar.

ACTITUDES Y VALORES A FOMENTAR

Puntualidad, responsabilidad, compromiso, disciplina.



METODOLOGÍA DE ENSEÑANZA APRENDIZAJE

Método	Método tradicional de exposición	Método Audiovisual	Aula Interactiva	Multimedia	Desarrollo de proyecto	Dinámicas	Estudio de casos	Otros (Especificar)
%	50	0	10	0	30	5	5	0

CONTENIDO TEMÁTICO

MODULO 1. Conceptos fundamentales sobre programación orientada a objetos.		29.6 HRS
<i>El alumno comprenderá los conceptos de la programación orientada a objetos, al emplearlos e identificarlos en escenarios propuestos en clase, aplicados en un lenguaje moderno orientado a objetos.</i>		
1.1	Introducción	1 HRS
<i>El alumno conocerá las herramientas de desarrollo a utilizar en el curso.</i>		
1.2	Abstracción	.3 HRS
<i>El alumno identificará y clasificará las propiedades de entidades propuestas con escenarios en clase, mediante el proceso de abstracción.</i>		
1.3	Objetos y clases	1 HRS
<i>El alumno identificará la diferencia y relación entre un objeto y su clase, al abstraer propiedades de entidades propuestas.</i>		
1.4	Encapsulamiento	.3 HRS
<i>El alumno determinará los métodos de interfaz necesario para lograr el encapsulamiento.</i>		
1.5	Comunicación entre objetos	1.5 HRS
<i>El alumno conocerá la forma de comunicar a los objetos que forman parte de escenarios propuestos en clase.</i>		
1.6	Generalización, especialización de clases	4 HRS
<i>El alumno conocerá la relación de generalización-especialización para crear relaciones jerárquicas.</i>		
1.6.1	Tipos de herencia	
<i>El alumno distinguirá las ventajas y desventajas entre herencia simple y herencia múltiple.</i>		
1.6.2	Clase abstracta	
<i>El alumno conocerá el concepto de clase abstracta para utilizarlo en los programas.</i>		



1.7	Expresiones del polimorfismo	7.5 HRS
	<i>El alumno aplicará el mecanismo del polimorfismo como una de las técnicas más útiles de la programación orientada a objetos.</i>	
1.7.1	Definición del polimorfismo	
	<i>El alumno conocerá el concepto de polimorfismo.</i>	
1.7.2	Sobre-carga de métodos	
	<i>El alumno aplicará la sobrecarga de métodos en sus programas.</i>	
1.7.3	Sobre-escritura de métodos	
	<i>El alumno aplicará la sobre-escritura de métodos en sus programas.</i>	
1.7.4	Método abstracto	
	<i>El alumno conocerá el concepto de método abstracto.</i>	
1.8	Relaciones entre clases	2 HRS
	<i>El alumno conocerá algunas relaciones que se aplican en los diagramas de clases.</i>	
1.8.1	Agregación y composición de clases	
	<i>El alumno aplicará distintos niveles de abstracción entre clases, comprobando que una entidad puede estar compuesta de otras al analizar las clases involucradas en contextos propuestos en clase.</i>	
1.8.2	Delegación	
	<i>El alumno identificará la delegación como método de repartición de tareas entre objetos para solucionar problemas.</i>	
1.9	Almacenamiento de objetos	12 HRS
	<i>El alumno utilizará los arreglos y algunas estructuras de datos para almacenar tipos de datos primitivos y objetos.</i>	
1.9.1	Almacenamiento de tipos primitivos en arreglos	
	<i>El alumno implementará programas donde se manejen arreglos de una dimensión y dos dimensiones.</i>	
1.9.2	Almacenamiento de objetos en arreglos	
	<i>El alumno implementará programas donde maneje arreglos de referencias para almacenar objetos.</i>	
1.9.3	Almacenamiento de objetos en arreglos dinámicos	
	<i>El alumno utilizará arreglos dinámicos en sus programas.</i>	
1.9.4	Listas	



		<i>El alumno implementara la estructura de datos lista.</i>	
1.9.5	Pilas		
		<i>El alumno implementara la estructura de datos pila.</i>	
1.9.6	Colas		
		<i>El alumno implementará la estructura de datos cola.</i>	
MODULO 2. Modelado e implementación de Objetos			7 HRS
<i>El alumno identificará las técnicas de modelado estandarizadas para el diseño de sistemas, y realizará la traducción del diseño a código en el lenguaje propuesto en clase.</i>			
2.1	Del Diseño al Código		7 HRS
		<i>El alumno aplicará el modelado en UML para codificar la solución de un problema.</i>	
MODULO 3. Objetos al servicio de la programación basada en componentes y dirigida por eventos.			15 HRS
<i>El alumno conocerá la aplicación de los objetos en la programación visual.</i>			
3.1	Construcción de Interfaces Gráficas		6.5 HRS
		<i>El alumno conocerá la aplicación de los objetos en la programación visual.</i>	
3.2	Interfaces Gráficas que responden a Eventos		8.5 HRS
		<i>El alumno diseñará interfaces gráficas que respondan a eventos, al implementar las acciones correspondientes a las peticiones del usuario.</i>	
MODULO 4. Flujos de Entrada y Salida			8.4 HRS
<i>El alumno conocerá los tipos básicos de entrada y salida de una aplicación.</i>			
4.1	Clase de flujo de entrada y salida básicas		6 HRS
		<i>El alumno identificará los distintos tipos de flujo a un programa.</i>	
4.2	Serialización de Objetos		2.4 HRS
		<i>El alumno estructurará métodos para obtener objetos persistentes, como solución a problemas que requieren información almacenada en memoria secundaria.</i>	



UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS
DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN



CRITERIOS DE EVALUACIÓN

60% prácticas
30% 2 exámenes parciales
10% tareas

BIBLIOGRAFÍA

BÁSICA

TITULO	AUTOR	EDITORIAL	AÑO DE EDICIÓN	% DE COBERTURA DEL CURSO
Programación con Java	Deitel & Deitel	Pearson	2008	100
Java 2, curso de programación, 3era. Edición	Fco. Javier Ceballos	Alfaomega	2006	100
Programación en Java	C. Thomas Wu	Mc Graw Hill	2008	100
Programación en Java 2	Luis Joyanes Aguilar	Mc Graw Hill	2002	100

COMPLEMENTARIA

TITULO	AUTOR	EDITORIAL	AÑO DE EDICIÓN	% DE COBERTURA DEL CURSO
El lenguaje unificado de modelado, 2da. Edición, UML 2.0	Grady Booch, James Rumbaugh, Ivar Jacobson	Pearson/Addison Wesley	2006	30
El lenguaje unificado de modelado (Manual de referencia), 2da. Edición, UML 2.0	James Rumbaugh, Ivar Jacobson, Grady Booch	Pearson/Addison Wesley	2007	30
Utilización de UML en Ingeniería del Software con objetos y componentes, 2da. Edición	Perdita Stevens, Rob Pooley	Pearson/Addison Wesley	2007	30

REVISIÓN REALIZADA POR:

NOMBRE DEL PROFESOR	FIRMA
Sabrina Vega Maldonado	
Violeta del Rocío Becerra Velázquez	
Lotzy Beatriz Fonseca Chiu	
Luis Alberto Casillas Santillán	
Luis Felipe Mariscal Lugo	
Marisela Mireles Mercado	
Luis Alberto Muñoz Gómez	



UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS
DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN



Vo.Bo. Presidente de Academia

Lic. Sabrina Lizberh Vega Maldonado

Vo.Bo. Jefe del Departamento

Dr. Marco Antonio Pérez Cisneros

miércoles, 15 de julio de 2009