



<p align="center"><b>UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</b></p>  <p align="center">Clave: 08MSU0017H</p> <p align="center"><b>FACULTAD DE INGENIERÍA</b></p>  <p align="center">Clave: 08USU4053W</p> <p align="center"><b>PROGRAMA ANALÍTICO DE LA UNIDAD DE APRENDIZAJE: PROGRAMACIÓN II</b></p>	<b>DES:</b>	Ingeniería
	<b>Programa(s) Educativo(s):</b>	Ingeniería de Software
	<b>Tipo de materia:</b>	Obligatoria
	<b>Clave de la materia:</b>	IS0603
	<b>Cuatrimestre:</b>	6
	<b>Área en plan de estudios:</b>	Específica
	<b>Créditos</b>	5.4
	<b>Total de horas por semana:</b>	4 horas
	<i>Teoría: Virtual</i>	4 horas
	<i>Práctica</i>	
	<i>Taller:</i>	
	<i>Laboratorio:</i>	
	<i>Prácticas complementarias:</i>	
	<i>Trabajo extra clase:</i>	4 horas
	<b>Total de horas por cuatrimestre:</b>	96 horas
<b>Fecha de actualización:</b>	Octubre de 2015	
<i>Materia requisito:</i>	IS0503 - Programación I	

**PROPÓSITO DEL CURSO:**

Introducir los principios y la práctica del desarrollo de software utilizando el paradigma orientado a objetos, mediante la aplicación y práctica con un lenguaje de programación de éste paradigma, para la solución de problemas algorítmicos desde un enfoque distinto.

COMPETENCIAS	DOMINIOS COGNITIVOS.	RESULTADOS DE APRENDIZAJE.
<p>Competencias específicas</p> <ul style="list-style-type: none"> <li><b>Sistemas computacionales:</b> Aplica los fundamentos de las Ciencias de la Computación que dan soporte al diseño y construcción de software mediante un proceso metodológico atendiendo a los estándares internacionales.</li> <li><b>Diseño y Modelado de Software:</b> Transforma los requerimientos del cliente en una especificación formal y documentada, diseñando y modelando soluciones de</li> </ul>	<ol style="list-style-type: none"> <li><b>Introducción al Paradigma Orientado a Objetos</b> <ol style="list-style-type: none"> <li>Clases</li> <li>Objetos</li> <li>Características del paradigma</li> <li>Modificadores de acceso public, private y protected</li> </ol> </li> <li><b>Métodos</b> <ol style="list-style-type: none"> <li>Declaración de métodos</li> <li>Concepto de método.</li> <li>Atributos const y static.</li> <li>Llamadas a métodos y uso de argumentos.</li> <li>Tipos de métodos.</li> <li>Autorreferencia</li> </ol> </li> </ol>	<ul style="list-style-type: none"> <li>✓ Identifica los principios del paradigma orientado a objetos para su implementación en el desarrollo de proyectos.</li> <li>✓ Diseña interfaces de comunicación humano-computadora tomando en cuenta las</li> </ul>

<p>técnicas y metodologías que responden a estándares internacionales.</p> <ul style="list-style-type: none"> <li>● <b>Calidad de Software:</b> Selecciona las técnicas adecuadas para asegurar la calidad y seguridad del software durante su planeación, diseño, construcción y mantenimiento mediante la aplicación de metodologías, técnicas y estándares internacionales, que aplicados sistemáticamente garanticen los requerimientos del cliente.</li> <li>● <b>Ingeniería del Proceso de Software:</b> Adquiere las herramientas y metodologías para llevar a cabo las etapas de la ingeniería del proceso de software para construir o mejorar proyectos mediante métodos y procedimientos para lograr su propósito.</li> </ul>	<p>2.7 Forma de pasar argumentos.</p> <p><b>3. Constructores y destructores</b></p> <p>3.1. Conceptos de métodos constructor y destructor.</p> <p>3.2. Declaración de métodos constructor y destructor.</p> <p>3.3. Aplicaciones de constructores y destructores.</p> <p>3.4. Tipos de constructores y destructores.</p> <p><b>4. Herencia</b></p> <p>5.1 Introducción a la herencia.</p> <p>5.2 Herencia simple.</p> <p>5.3 Herencia múltiple.</p> <p>5.4 Clase base y clase derivada.</p> <p>5.5 Parte protegida.</p> <p>5.6 Redefinición de los miembros de las clases derivadas.</p> <p>5.7 Clases virtuales y visibilidad.</p> <p>5.8 Constructores y destructores en clases derivadas.</p> <p>5.9 Aplicaciones.</p> <p><b>5. Polimorfismo y reutilización</b></p> <p>5.1. Concepto del polimorfismo.</p> <p>5.2. Clases abstractas.</p> <p>5.3. Definición de una interfaz.</p> <p>5.4. Implementación de la definición de una interfaz.</p> <p>5.5. Reutilización de la definición de una interfaz.</p> <p>5.6. Definición y creación de paquetes / librería.</p> <p>5.7. Reutilización de las clases de un paquete / librería.</p> <p>5.8. Clases genéricas (Plantillas).</p> <p><b>6. Excepciones</b></p> <p>6.1. Propagación</p> <p>6.2. Gestión de Excepciones</p>	<p>necesidades de pertenencia de las personas.</p> <ul style="list-style-type: none"> <li>✓ Utiliza la notación formal de diseño describiendo de manera precisa lo que el proyecto de software debe hacer.</li> <li>✓ Utiliza métricas en la toma de decisiones durante el proceso de ingeniería de software.</li> <li>✓ Implementa métodos y estrategias coadyuvantes a la construcción de software seguro, garantizando su calidad.</li> </ul>
--	---	--

OBJETOS DE APRENDIZAJE	METODOLOGÍA (Estrategias, secuencias, recursos didácticos)	EVIDENCIAS DE APRENDIZAJE.
<ol style="list-style-type: none"> <li>1. <b>Introducción al Paradigma Orientado a Objetos</b></li> <li>2. <b>Métodos</b></li> <li>3. <b>Constructores y destructores</b></li> <li>4. <b>Herencia</b></li> <li>5. <b>Polimorfismo y reutilización</b></li> <li>6. <b>Excepciones</b></li> </ol>	<ol style="list-style-type: none"> <li>1. Foro de Bienvenida, Documentos de lectura, Videos.</li> <li>2. Videos y documentos de lectura.</li> <li>3. Videos y documentos de lectura.</li> <li>4. Videos y documentos de lectura.</li> <li>5. Videos y documentos de lectura.</li> <li>6. Videos y documentos de lectura.</li> </ol>	<ol style="list-style-type: none"> <li>1. Elaboración de resúmenes, tareas y cuestionarios.</li> <li>2. Elaboración de tareas y ejercicios en línea.</li> <li>3. Elaboración de tareas y ejercicios en línea.</li> <li>4. Elaboración de tareas y ejercicios en línea.</li> <li>5. Elaboración de tareas y ejercicios en línea.</li> <li>6. Elaboración de tareas en línea.</li> </ol>
FUENTES DE INFORMACIÓN (Bibliografía, direcciones electrónicas)		EVALUACIÓN DE LOS APRENDIZAJES (Criterios e instrumentos)
<p>Peter Coad, Object Oriented Programming, Yourdon Press Computer Series, 1993.</p> <p>Robert Lafore, Object Oriented Programming in C++, 2001</p>		<p>Se toma en cuenta para integrar la calificación total:</p> <p><b>Introducción al Paradigma Orientado a Objeto</b></p> <ul style="list-style-type: none"> <li>● Cuestionario sobre Clases y Objetos: <b>5%</b></li> <li>● Cuestionario sobre el Paradigma: <b>10%</b></li> <li>● Tarea sobre modificadores de acceso: <b>10%</b></li> </ul> <p><b>Métodos</b></p> <ul style="list-style-type: none"> <li>● Ejercicios con métodos: <b>10%</b></li> <li>● Ejercicios con métodos auto referenciados: <b>5%</b></li> </ul> <p><b>Constructores y Destructores</b></p> <ul style="list-style-type: none"> <li>● Tarea de investigación de Constructores y Destructores: <b>10%</b></li> <li>● Ejercicios con Constructores y Destructores: <b>10%</b></li> </ul>

