


<p style="text-align: center;"><b>UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</b></p>  <p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE <b>CHIHUAHUA</b></p> <p style="text-align: center;"><b>PROGRAMA ANALÍTICO DE LA UNIDAD DE APRENDIZAJE:</b></p> <p style="text-align: center;"><b>PROGRAMACION ORIENTADA A OBJETOS</b></p>	<b>DES:</b>	<b>INGENIERÍA</b>
	<b>Programa Educativo</b>	Ingeniería en Ciencias de la Computación
	<b>Tipo de materia (Obli/Opta):</b>	Obligatoria
	<b>Clave de la materia:</b>	CI475
	<b>Semestre:</b>	4
	<b>Área en plan de estudios (G, E):</b>	Ciencias de la Ingeniería
	<b>Total, de horas por semana:</b>	4
	<i>Teoría: Presencial o Virtual</i>	3
	<i>Laboratorio o Taller:</i>	1
	<i>Prácticas:</i>	0
	<i>Trabajo extra-clase:</i>	0
	<b>Créditos Totales:</b>	4
	<b>Total, de horas semestre (x 16 sem):</b>	64
	Fecha de actualización:	Febrero 2023
	<i>Prerrequisito (s):</i>	Ninguno
<i>Realizado por:</i>	Comité de Rediseño Curricular	

**PROPÓSITO DEL CURSO**

El curso le proporciona al estudiante los conceptos y técnicas para que sea capaz de escribir programas simples utilizando el paradigma orientado a objetos. El curso lo introduce a la definición y aplicación de los conceptos de herencia, abstracción, encapsulamiento, interfaces, polimorfismo, colecciones, excepciones y concurrencia, generando programas más entendibles, simplificados, de fácil mantenimiento y actualización, con reutilización de código y protección de datos.

COMPETENCIAS (Tipo Y Nombre de la competencias que nutre la materia y a las que contribuye)	DOMINIOS COGNITIVOS (Objetos de estudio, temas y subtemas)	RESULTADOS DE APRENDIZAJE. (Por objeto de estudio).

<p>El curso promueve las siguientes competencias:</p> <p><b>BÁSICAS:</b></p> <p><b>SOLUCIÓN DE PROBLEMAS</b></p> <ul style="list-style-type: none"> <li>• Aplica las diferentes técnicas de observación para la solución de problemas.</li> <li>• Distingue los diversos tipos de sistemas</li> <li>• Aplica la tecnología a la solución de problemáticas</li> </ul>	<p><b>I. INTRODUCCION A LA PROGRAMACION ORIENTADA A OBJETOS</b></p> <ol style="list-style-type: none"> <li>1.1. Breve historia de los paradigmas de programación</li> <li>1.2. Comparación entre paradigmas.</li> <li>1.3. Fundamentos de la programación orientada a objetos.</li> <li>1.4. Conceptos básicos de la programación orientada a objetos <ol style="list-style-type: none"> <li>1.4.1.Clase</li> <li>1.4.2.Objeto</li> <li>1.4.3.Método</li> <li>1.4.4.Evento</li> <li>1.4.5.Programa</li> <li>1.4.6.Paquete</li> </ol> </li> </ol>	<p>Define y analiza los conceptos básicos de la teoría de objetos en el paradigma de programación orientado a objetos.</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------

<ul style="list-style-type: none"> <li>• Aplica el enfoque sistémico en diversos contextos</li> <li>• Crea soluciones innovadoras y utiliza formas no convencionales en la solución de problemas</li> </ul>	<p>1.4.7. Propiedad</p> <p>1.5. Principios fundamentales de la orientación a objetos.</p> <p>1.5.1. Abstracción</p> <p>1.5.2. Encapsulamiento</p> <p>1.5.3. Herencia</p> <p>1.6. Principios extras</p> <p>1.6.1. Polimorfismo</p> <p>1.6.2. Concurrencia</p> <p>1.6.3. Persistencia</p>	
<p><b>PROFESIONALES:</b></p> <p><b>CIENCIAS FUNDAMENTALES DE LA INGENIERÍA</b></p> <ul style="list-style-type: none"> <li>• Utiliza las matemáticas como herramientas para solución de problemas en ingeniería</li> </ul>	<p><b>II. HERENCIA, POLIMORFISMO E INTERFACES</b></p> <p>2.1 Herencia</p> <p>2.1.1. Jerarquía de clases</p> <p>2.1.1.1 Súper Clases</p> <p>2.1.1.2 Sub Clases</p> <p>2.1.2 Derivación de propiedades y constructores de clases</p> <p>2.1.3 Derivación de métodos</p> <p>2.1.4 Herencia simple y múltiple</p> <p>2.2 Polimorfismo</p> <p>2.2.1 Métodos polimórficos</p> <p>2.2.2 Sobrecarga de operadores</p> <p>2.2.2.1 Extensión de clases existentes</p> <p>2.2.2.2 Conversiones de tipos de datos</p> <p>2.3 Clases abstractas</p> <p>2.4 Interfaces</p> <p>2.4.1 Interfaces estándar para añadir Funcionalidad</p> <p>2.4.2 Creación de nuevas interfaces</p>	<p>Enumera y explica los conceptos de herencia, polimorfismo e interfaces en la programación orientada a objetos.</p> <p>Emplea los conocimientos sobre herencia polimorfismo e interfaces para escribir programas simples orientados a objetos.</p>
<p><b>ESPECIFICAS:</b></p> <p><b>FUNDAMENTOS DE CIENCIAS DE LA COMPUTACIÓN</b></p> <ul style="list-style-type: none"> <li>• Diseña y aplica algoritmos, estructuras y representación de datos para soluciones computacionales</li> <li>• Distingue los fundamentos de un sistema computacional</li> <li>• Aplica las bases de los lenguajes de programación para generar aplicaciones óptimas</li> <li>• Aplica los tópicos de paradigmas de</li> </ul>	<p><b>III. COLECCIONES</b></p> <p>3.1 Listas (Lists)</p> <p>3.2 Conjuntos (Sets)</p> <p>3.3 Mapas (Maps)</p> <p>3.4 Métodos de ordenamiento</p> <p>3.4 Implementación de colecciones</p>	<p>Define el concepto de colección en el ámbito de la programación orientada a objetos</p> <p>Desarrolla programas simples empleando el concepto de colección en la programación orientada a objetos</p>

<p>programación: estructurado, orientado a objetos. Lógico, funcional, entre otros para la creación de aplicaciones óptimas</p>		
	<p><b>IV. MANEJO DE EXCEPCIONES</b></p> <p>4.1 Tipos de excepciones  4.2 "Lanzando" Excepciones (Throwing Exceptions)  4.3 "Atrapando" Excepciones (Catching Exceptions)  4.4 Extensión de la Clase "Exception"</p>	<p>Ilustra los tipos de excepciones de un programa en el paradigma orientado a objetos.</p> <p>Utiliza el concepto de excepción para escribir programas robustos orientados a objetos</p>
	<p><b>V. CONCURRENCIA</b></p> <p>5.1 Procesos e hilos de control  5.1.1 Hilos  5.1.2 Juntas de hilos  5.2 Sincronización  5.2.1 Locks y bloques sincronizados</p>	<p>Define el concepto de concurrencia en la programación orientada a objetos.</p> <p>Desarrolla programas de cómputo concurrente orientados a objetos.</p>

<b>OBJETO DE ESTUDIO</b>	<b>METODOLOGIA</b> (Estrategias, secuencias, recursos didácticos)	<b>EVIDENCIAS DE APRENDIZAJE.</b>
<p>I. INTRODUCCION A LA PROGRAMACION ORIENTADA A OBJETOS</p> <p>II. HERENCIA</p> <p>III. POLIMORFISMO E INTERFACES</p> <p>IV. COLECCIONES</p> <p>V. MANEJO DE EXCEPCIONES</p> <p>V. CONCURRENCIA</p>	<p>Aprendizaje interactivo (exposición del profesor)</p> <p>Grupo de discusión.</p> <p>Auto aprendizaje (búsqueda y análisis de información)</p> <p>Inductivo</p> <ul style="list-style-type: none"> <li>• Observación</li> <li>• Comparación</li> </ul> <p>Deductivo</p> <ul style="list-style-type: none"> <li>• Aplicación</li> <li>• Comprobación</li> <li>• Demostración</li> </ul> <p>Sintético</p> <ul style="list-style-type: none"> <li>• Recapitulación</li> <li>• Definición</li> <li>• Resumen</li> <li>• Esquemas</li> <li>• Modelos matemáticos</li> <li>• Conclusión.</li> </ul>	<ul style="list-style-type: none"> <li>• Tareas y ejercicios.</li> <li>• Escritura de programas de computo.</li> <li>• Exámenes escritos.</li> </ul>

	<p><b>Material de Apoyo didáctico: Recursos</b></p> <ul style="list-style-type: none"> <li>• Manual de Instrucción</li> <li>• Materiales gráficos: artículos, libros, diccionarios, etc.</li> <li>• Cañón</li> <li>• Pintarrones</li> <li>• Equipo de computo</li> </ul>	
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

<b>FUENTES DE INFORMACIÓN</b> (Bibliografía, Direcciones electrónicas)	<b>EVALUACIÓN DE LOS APRENDIZAJES</b> (Criterios e instrumentos)
<ol style="list-style-type: none"> <li>1. Concepts of Programming Languages. Robert W. Sebesta. Cualquier edición.</li> <li>2. Como programar en Java. Deitel. Pearson - Prentice Hall.</li> <li>3. Como programar en C++. Deitel. Pearson - Prentice Hall.</li> </ol>	<p><b>INSTRUMENTOS:</b></p> <ul style="list-style-type: none"> <li>• Exámenes</li> <li>• Solución de ejercicios (aplicación de conocimientos)</li> <li>• Lista de cotejo (Respeto y participación al trabajo dentro del salón de clase, interés por la asignatura)</li> </ul> <p><b>CRITERIOS DE DESEMPEÑO:</b></p> <ul style="list-style-type: none"> <li>• Los programas desarrollados deberán estar completos.</li> <li>• Las tareas se deberán entregar en tiempo y forma</li> </ul> <p style="margin-left: 40px;">• <b>Exámenes escritos:</b></p> <p>Se realizan 3 exámenes escritos durante el semestre y las fechas se establecen por la secretaría académica</p> <p>Se toma en cuenta para integrar <b>calificaciones parciales:</b></p> <p>Exámenes 60% Tareas y practicas 40%</p> <p>Fecha de exámenes parciales: 1º. Parcial: 2º. Parcial: 3º. Parcial:</p> <p><b>LAS ACTIVIDADES NO REALIZADAS EN TIEMPO Y FORMA SE CALIFICAN CON CERO.</b></p> <p><b>Nota:</b> para acreditar el curso se deberá tener calificación aprobatoria tanto en la teoría como en las prácticas.</p>

Cronograma del Avance Programático

S e m a n a s

Unidades de aprendizaje	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I. Introducción a la programación orientada a objetos	■	■														
II. Herencia, polimorfismo e interfaces			■	■	■	■	■	■								
III. Colecciones									■	■	■					
IV. Manejo de excepciones												■	■			
V. Concurrencia														■	■	■