

## 1. Datos Generales de la asignatura

<b>Nombre de la asignatura:</b>	Programación Orientada a Objetos
<b>Clave de la asignatura:</b>	AEB-1054
<b>SATCA<sup>1</sup>:</b>	1-4-5
<b>Carrera:</b>	Ingeniería Informática e Ingeniería en Tecnologías de la Información y Comunicaciones

## 2. Presentación

### Caracterización de la asignatura

Esta asignatura aporta al perfil del egresado habilidades para analizar, modelar desarrollar e implementar sistemas de información para la gestión de procesos y apoyo en la toma de decisiones que aumenten la productividad y competitividad de las organizaciones utilizando metodologías basadas en estándares internacionales.

La asignatura incluye los conceptos y metodología orientada a objetos que permiten la creación y optimización de aplicaciones basadas en clases y objetos, que propicien un desarrollo ágil a través de la reutilización de código y aplicación de control de errores.

Para cursar esta asignatura se requiere de conocimientos básicos y lógicos de programación como las reglas de diagramación, pseudocódigo, uso de expresiones de declaración, asignación, condicionales y estructuras cíclicas que son adquiridos en la asignatura de Fundamentos de Programación.

Se relaciona con todas asignaturas posteriores en donde se apliquen metodologías de programación.

### Intención didáctica

Este programa de estudios se sugiere eminentemente práctico, es decir, el docente propone problemáticas y el estudiante deberá resolverlas, aplicando el paradigma orientado a objetos. El contenido está organizado en seis temas.

El primer tema introduce al estudiante en un entorno de desarrollo de programación.

En los temas dos, tres y cuatro se tratan los conceptos de la programación orientada a objetos.

El tema cinco aborda el tratamiento de arreglos y en el tema seis se trata lo referente a archivos.

Como esta asignatura se centra en la solución de problemas en un lenguaje de programación orientado a objetos, se recomienda desarrollar programas demostrativos en cada tema, aplicando los conceptos vistos en clase y poniendo atención en el avance de los estudiantes.

<sup>1</sup> Sistema de Asignación y Transferencia de Créditos Académicos

### 3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Evento
<p>Instituto Tecnológico de Aguascalientes del 15 al 18 de junio de 2010.</p>	<p>Representantes de los Institutos Tecnológicos de:            Centro Interdisciplinario de Investigación y Docencia en Educación Técnica, Acapulco, Aguascalientes, Apizaco, Boca Río, Celaya, Chetumal, Chihuahua, Chilpancingo, Chiná, Cd. Cuauhtémoc, Cd. Juárez, Cd. Madero, Cd. Victoria, Colima, Comitán, Cautla, Durango, El Llano de Aguascalientes, Huixquilucan, Valle Bravo, Guaymas, Huatabampo, Huejutla, Iguala, La Laguna, La Paz, La Zona Maya, León, Lerma, Linares, Los Mochis, Matamoros, Mazatlán, Mérida, Mexicali, Minatitlán, Nuevo Laredo, Orizaba, Pachuca, Puebla, Querétaro, Reynosa, Roque, Salina Cruz, Saltillo, San Luis Potosí, Tehuacán, Tepic, Tijuana, Tlaxiaco, Toluca, Torreón, Tuxtepec, Valle de Oaxaca, Veracruz, Villahermosa, Zacatecas, Zacatepec, Altiplano de Tlaxcala, Coatzacoalcos, Cuautitlán Izcalli, Fresnillo, Irapuato, La Sierra Norte Puebla, Macuspana, Naranjos, Pátzcuaro, Poza Rica, Progreso, Puerto Vallarta, Tacámbaro, Tamazula Gordiano, Tlaxco, Venustiano Carranza, Zacapoaxtla, Zongólica y Oriente del Estado Hidalgo.</p>	<p>Elaboración del programa de estudio equivalente en la Reunión Nacional de Implementación Curricular y Fortalecimiento Curricular de las asignaturas comunes por área de conocimiento para los planes de estudio actualizados del SNEST.</p>
<p>Instituto Tecnológico de Morelia del 10 al 13 de septiembre de 2013.</p>	<p>Representantes de los Institutos Tecnológicos de:            Aguascalientes, Apizaco, Boca del Río, Celaya, CRODE</p>	<p>Reunión Nacional de Seguimiento Curricular de las Asignaturas Equivalentes del SNIT.</p>

	<p>Celaya, Cerro Azul, Chihuahua, Cd. Cuauhtémoc, Cd. Hidalgo, Cd. Juárez, Cd. Madero, Cd. Valles, Coacalco, Colima, Iguala, La Laguna, Lerdo, Los Cabos, Matamoros, Mérida, Morelia, Motúl, Múzquiz, Nuevo Laredo, Nuevo León, Oriente del Estado de México, Orizaba, Pachuca, Progreso, Purhepecha, Salvatierra, San Juan del Río, Santiago Papasquiario, Tantoyuca, Tepic, Tlatlauquitpec, Valle de Morelia, Venustiano Carranza, Veracruz, Villahermosa, Zacatecas y Zacatepec.</p>	
--	---	--

#### 4. Competencia(s) a desarrollar

<b>Competencia(s) específica(s) de la asignatura</b>
<p>Aplica el paradigma orientado a objetos para el desarrollo de aplicaciones que solucionen problemas del entorno.</p>

#### 5. Competencias previas

<ul style="list-style-type: none"> <li>• Aplica las herramientas básicas de programación orientada a objetos, para modelar y desarrollar soluciones a diversos problemas del mundo real.</li> </ul>
---

#### 6. Temario

No.	Temas	Subtemas
1	Entorno de desarrollo	1.1 Estructura de un proyecto 1.2 Proceso de desarrollo de aplicaciones 1.2.1 Compilación 1.2.2 Archivos que crea el entorno de desarrollo 1.2.3 Depuración
2	Clases y objetos	2.1 Definición de una clase 2.2 Declaración de clases 2.3 Miembros de una clase 2.4 Ámbito referente a una clase 2.5 Especificadores de acceso 2.6 Creación de objetos 2.7 Clases predefinidas 2.8 Definición, creación y reutilización de paquetes/librerías 2.9 Manejo de excepciones

3	Métodos	<p>3.1 Definición de un método. 3.2 Estructura de un método. 3.3 Valor de retorno. 3.4 Declaración de un método. 3.4.1 De clase 3.4.2 De Instancia 3.5 Ámbito y tiempo de vida de variables. 3.6 Argumentos y paso de parámetros 3.7 Puntero this. 3.8 Sobrecarga de métodos. 3.9 Constructores y destructores</p>
4	Herencia y polimorfismo	<p>4.1 Concepto de herencia y polimorfismo. 4.2 Definición de una clase base. 4.3 Definición de una clase derivada. 4.4 Clases abstractas. 4.5 Interfaces. 4.5.1 Definición 4.5.2 Implementación. 4.5.3 Variable polimórficas 4.6 Reutilización de la definición de paquetes/librerías</p>
5	Arreglos	<p>5.1 Unidimensional 5.2 Multidimensional</p>
6	Archivos	<p>6.1 Definición de Archivos de texto y archivos binarios. 6.2 Operaciones básicas en archivos texto y binario. 6.3 Manejo de excepciones en archivos.</p>

## 7. Actividades de aprendizaje de los temas

Entorno de desarrollo	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> Conoce y aplica la estructura de proyectos en un entorno de desarrollo para la solución de problemas.</p> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>Comunicación oral y escrita</li> <li>Habilidades del manejo de la computadora</li> <li>Capacidad de aplicar los conocimientos en la práctica</li> <li>Habilidad para trabajar en forma autónoma</li> </ul>	<ul style="list-style-type: none"> <li>Reconocer la estructura de un proyecto de programación en un entorno de desarrollo a través de la elaboración de un esquema gráfico.</li> <li>Aplicar las fases del proceso de desarrollo de aplicaciones en la resolución de ejercicios</li> <li>Identificar los archivos que se generan en los proyectos creados y presentar un cuadro sinóptico.</li> <li>Utilizar la herramienta de depuración del entorno de programación para identificar y corregir errores lógicos y registrarlo en una bitácora.</li> </ul>

Clases y objetos	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> Comprende y aplica la estructura de clases para la creación de objetos y utiliza clases predefinidas para facilitar el desarrollo de aplicaciones</p> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>• Solución de problemas</li> <li>• Capacidad de aplicar los conocimientos en la práctica</li> <li>• Capacidad de aprender</li> <li>• Capacidad de generar nuevas ideas (creatividad)</li> <li>• Habilidad para trabajar en forma autónoma.</li> </ul>	<ul style="list-style-type: none"> <li>• Crear aplicaciones que solucionen problemas planteados por el docente que incluyan todos los conceptos del tema de estudio, entregar informes</li> <li>• Identificar los componentes de una clase y elaborar un esquema gráfico donde se describan.</li> <li>• Identificar y crear clases que ameriten su reutilización y a partir de ellas definir paquetes, elaborar el reporte.</li> </ul>
Métodos	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> Comprende y aplica los diferentes tipos de métodos, tomando en cuenta el ámbito y tiempo de vida de los datos durante la ejecución de un programa.</p> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis</li> <li>• Habilidad para buscar y analizar información proveniente de fuentes diversas</li> <li>• Solución de problemas</li> <li>• Capacidad de aplicar los conocimientos en la práctica</li> <li>• Capacidad de aprender</li> <li>• Habilidad para trabajar en forma autónoma</li> </ul>	<ul style="list-style-type: none"> <li>• Investigar el concepto de método y definir su estructura en un esquema gráfico.</li> <li>• Construir clases que solucionen problemas planteados por el docente utilizando métodos de cualquier tipo, entregar reporte.</li> <li>• Crear aplicaciones en donde el alumno utilice diferentes tipos de variable, argumentos y paso de parámetros, así como diferentes métodos constructores y destructores para la creación y eliminación de objetos en diversas clases observando y registrando sus conclusiones a través de un reporte.</li> <li>• Crear aplicaciones donde se involucren clases que hagan uso de sobrecarga de métodos y exponer su funcionamiento.</li> </ul>
Herencia y polimorfismo	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> Comprende y aplica los conceptos de herencia y polimorfismo en programas que utilicen clases base, clases derivadas, clases abstractas e interfaces.</p> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis</li> <li>• Habilidad para buscar y analizar</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar las diferentes clases que existen, así como los conceptos de herencia y polimorfismo y presentar un mapa mental.</li> <li>• Desarrollar aplicaciones que implementen el concepto de herencia a partir de un modelo de clases que la utilice al menos una clase base y al menos una clase derivada, exponiendo los códigos generados.</li> <li>• Programar clases abstractas para la solución de</li> </ul>



información proveniente de fuentes diversas <ul style="list-style-type: none"> <li>• Solución de problemas</li> <li>• Capacidad de aplicar los conocimientos en la práctica</li> <li>• Habilidad para trabajar en forma autónoma</li> <li>• Capacidad de aprender</li> </ul>	un problema utilizando clases derivadas para la implementación de métodos abstractos de la clase base, entregando informes y códigos. <ul style="list-style-type: none"> <li>• Crear aplicaciones que implementen polimorfismo con el uso de interfaces, entregando el código.</li> </ul>
<b>Arreglos</b>	
<b>Competencias</b>	<b>Actividades de aprendizaje</b>
<p><b>Específica(s):</b>                  Conoce y aplica programas que implementen el uso de arreglos para reconocerlos como una herramienta para agrupar datos y facilitar la solución de problemas.</p> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis</li> <li>• Habilidad para buscar y analizar información proveniente de fuentes diversas</li> <li>• Solución de problemas</li> <li>• Trabajo en equipo</li> <li>• Capacidad de aplicar los conocimientos en la práctica</li> <li>• Capacidad de aprender.</li> </ul>	<ul style="list-style-type: none"> <li>• Investigar los conceptos de arreglo unidimensional y multidimensional y presentar una síntesis.</li> <li>• Resolver ejercicios donde se identifiquen los elementos principales de los arreglos.</li> <li>• Crear aplicaciones que usen arreglos de diferentes dimensiones en la solución de problemas entregando los códigos.</li> </ul>
<b>Archivos</b>	
<b>Competencias</b>	<b>Actividades de aprendizaje</b>
<p><b>Específica(s):</b>                  Comprende y aplica programas que manipulen archivos de texto y binarios en función de una problemática dada.</p> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis</li> <li>• Solución de problemas</li> <li>• Habilidad para trabajar en forma autónoma</li> <li>• Capacidad de aplicar los conocimientos en la práctica</li> <li>• Capacidad de aprender</li> </ul>	<ul style="list-style-type: none"> <li>• Elaborar un cuadro comparativo con las características de los archivos de texto y binarios, así como su aplicación.</li> <li>• Desarrollar aplicaciones que usen archivos binarios que validen excepciones para la solución de problemas.</li> <li>• Documentar las prácticas</li> </ul>

## 8. Práctica(s)

- Realizar programas de conversión de unidades métricas, monedas, tablas de multiplicar, en función a una calificación imprimir un mensaje y otros.
- Realizar programas en donde se construyan clases, sobre problemas dados y relacionados con otras asignaturas, la definición de los atributos y la construcción de los objetos.
- Realizar programas en donde construyan las clases de problemas dados y relacionados con otras asignaturas, la definición de los métodos (vacíos o con parámetros, regresen o no valores) y la construcción de los objetos.
- Realizar programas en donde construyan las clases utilizando la herencia y apliquen el concepto de polimorfismo.
- Realizar programas de ordenamiento de números en un arreglo unidimensional o bidimensional utilizando los diversos métodos, resolver la determinante de una matriz, etc.
- Realizar programas donde los datos se almacenen en archivos ya sea de texto o binarios, ejemplo registro de clientes, registros de inventario, un editor, etc.

## 9. Proyecto de asignatura

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.
- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

## 10. Evaluación por competencias

Para evaluar las actividades de aprendizaje se recomienda solicitar: reporte de investigación, reportes de prácticas, estudios de casos, exposiciones en clase, cuadro comparativo, ejercicios, etc.

Para verificar el nivel del logro de las competencias del estudiante se recomienda utilizar: listas de cotejo, listas de verificación, matrices de valoración, guías de observación, rúbricas, etc.

## 11. Fuentes de información

1. Ceballos J. (2007) *Java 2 Lenguaje y aplicaciones*. España: Alfaomega.
2. Ceballos J. (2012) *Microsoft C# -Curso de Programación*. España: Alfaomega.
3. Dean J. y Dean R. (2009) *Introducción a la programación con Java*: McGraw Hill
4. Doyle, B (2013) *C# Programming: From Problem Analysis to Program Design*. Cengage Learning
5. Eckel, B. (2009) *Piensa en Java*. España: Prentice Hall.
6. Froufe, A. (2008) *Java 2 Manual de usuario y tutorial*. España: Alfaomega
7. Groussard, T. (2011) *Recursos Informáticos C#4 Los fundamentos del lenguaje- desarrollar con visual estudio 2010*. España: Eni Ediciones
8. Joyanes, L. y Zahonero, I. (2011) *Programación en Java 6*. España: McGraw Hill
9. Sierra, K. (2008) *SCJP Sun Certified Programmer for Java 6*. USA: McGraw Hill.
10. Wu T. (2008) *Programación en Java. Introducción a la Programación Orientada a Objetos*. España: McGrawHill