

1.- DATOS DE LA ASIGNATURA

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

2.- PRESENTACIÓN

Caracterización de la asignatura.

En esta asignatura aporta al perfil del egresado los conceptos de la metodología orientada a objetos que permiten la creación y optimización de aplicaciones basadas en clases. Se estudian aquí los mecanismos que permitan un desarrollo ágil a través de la reutilización de tipos de objetos y control de errores.

Intención didáctica.

El temario está organizado en seis unidades. Las primeras dos unidades, están centradas en los conceptos básico de los lenguajes de programación; Las unidades tres, cuatro y cinco, se tratan los conceptos de la programación orientada a objetos y en la unidad seis, lo referente a archivos.

La importancia de la materia se centra en la solución de problemas en un lenguaje de programación orientado a objetos, por lo que se recomienda desarrollar programas demostrativos en cada unidad de la aplicación de los conceptos vistos en clase, poniendo atención en los avances de los estudiantes.

¹ Sistema de Asignación y Transferencia de Créditos Académicos

3.- COMPETENCIAS A DESARROLLAR

<p>Competencias específicas:</p> <ul style="list-style-type: none">▪ Utilizar técnicas de modelado para la solución de problemas.▪ Aplicar la sintaxis de un lenguaje orientado a objetos.▪ Aplicar un lenguaje orientado a objetos para la solución de problemas.	<p>Competencias genéricas:</p> <p><u>Competencias instrumentales</u></p> <ul style="list-style-type: none">• Capacidad de análisis y síntesis.• Capacidad de organizar y planificar.• Conocimientos básicos de la carrera.• Comunicación oral y escrita.• Habilidades del manejo de la computadora.• Habilidad para buscar y analizar información proveniente de fuentes diversas.• Solución de problemas.• Toma de decisiones. <p><u>Competencias interpersonales</u></p> <ul style="list-style-type: none">• Capacidad crítica y autocrítica.• Trabajo en equipo.• Habilidades interpersonales. <p><u>Competencias sistémicas</u></p> <ul style="list-style-type: none">• Capacidad de aplicar los conocimientos en la práctica.• Habilidades de investigación.• Capacidad de aprender.• Capacidad de generar nuevas ideas (creatividad).• Habilidad para trabajar en forma autónoma.• Búsqueda del logro.	
---	---	--

4.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Evento
<p>Instituto Tecnológico Superior de Puerto Vallarta del 10 al 14 de agosto de 2009.</p>	<p>Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Superior de Centla, Chetumal, Ciudad Cuauhtémoc, Ciudad Madero, Comitán, Delicias, León, Superior de Misantla, Pachuca, Pinotepa, Puebla, Superior de Puerto Vallarta, Roque, Tepic, Tijuana, Tuxtla Gutiérrez y Villahermosa.</p>	<p>Reunión Nacional de Diseño e Innovación Curricular para el Desarrollo y Formación de Competencias Profesionales de la Carrera de Ingeniería en Tecnologías de la Información y Comunicaciones.</p>
<p>Desarrollo de Programas en Competencias Profesionales por los Institutos Tecnológicos del 17 de agosto de 2009 al 21 de mayo de 2010.</p>	<p>Academias de Ingeniería en Tecnologías de la Información y Comunicaciones de los Institutos Tecnológicos: Roque, Tepic y Villahermosa.</p>	<p>Elaboración del programa de estudio propuesto en la Reunión Nacional de Diseño Curricular de la Carrera de Ingeniería en Tecnologías de la Información y Comunicaciones.</p>
<p>Instituto Tecnológico de Villahermosa del 24 al 28 de mayo de 2010.</p>	<p>Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Superior de Centla, Chetumal, León, Pachuca, Puebla, Roque, Tepic, Tuxtla Gutiérrez y Villahermosa.</p>	<p>Reunión Nacional de Consolidación de los Programas en Competencias Profesionales de la Carrera de Ingeniería en Tecnologías de la Información y Comunicaciones.</p>
<p>Instituto Tecnológico Saltillo, del 5 al 9 de octubre de 2009.</p>	<p>Representantes de los Institutos Tecnológicos de: Apizaco, Cerro Azul, Chetumal, Ciudad Juárez, Ciudad Madero, Superior de Coahuila de Zaragoza, Colima, Comitancillo, Conkal, Durango, El Llano, Aguascalientes, El Salto, Superior de Fresnillo, Huejutla, Superior de Lerdo, Linares, Los Mochis, Mexicali, Morelia, Oaxaca, Superior del Occidente del Estado de Hidalgo, Ocotlán, Orizaba, Piedras Negras, Pinotepa, Saltillo, San Luis Potosí, Tapachula, Tijuana, Torreón, Tuxtepec, Superior de Valladolid, Valle del Guadiana, Superior de Zacapoaxtla y Zacatecas.</p>	<p>Reunión Nacional de Diseño e Innovación Curricular para el Desarrollo y Formación de Competencias Profesionales de la Carrera de Ingeniería Informática.</p>

Lugar y fecha de elaboración o revisión	Participantes	Evento
Desarrollo de Programas en Competencias Profesionales por los Institutos Tecnológicos del 12 de octubre de 2009 al 19 de febrero de 2010.	Academias de Ingeniería Informática de los Institutos Tecnológicos: Apizaco, Ciudad Juárez, Los Mochis, Oaxaca, Pinotepa, Tijuana, Torreón, Zacatecas, Superior de Lerdo y Superior de Zacapoaxtla.	Elaboración del programa de estudio propuesto en la Reunión Nacional de Diseño Curricular de la Carrera de Ingeniería Informática.
Instituto Tecnológico Superior de Poza Rica, del 22 al 26 de febrero de 2010.	Representantes de los Institutos Tecnológicos de: Apizaco, Cerro Azul, Chetumal, Ciudad Juárez, Ciudad Madero, Superior de Coatzacoalcos, Colima, Comitancillo, Conkal, Durango, El Llano Aguascalientes, El Salto, Superior de Fresnillo, Huejutla, Superior de Lerdo, Los Mochis, Mexicali, Morelia, Oaxaca, Superior del Occidente del Estado de Hidalgo, Ocotlán, Orizaba, Piedras Negras, Pinotepa, Saltillo, San Luis Potosí, Tapachula, Tijuana, Torreón, Tuxtepec, Superior de Valladolid, Valle del Guadiana, Superior de Zacapoaxtla y Zacatecas.	Reunión Nacional de Consolidación de los Programas en Competencias Profesionales de la Carrera de Ingeniería Informática.
Instituto Tecnológico de Aguascalientes, del 15 al 18 de Junio de 2010.	Representantes de los Institutos Tecnológicos de: Acapulco, Aguascalientes, Altiplano de Tlaxcala, Apizaco, Boca del Río, Ciudad Cuauhtémoc, Ciudad Juárez, Ciudad Madero, Ciudad Victoria, Celaya, Chetumal, Chihuahua, Chilpancingo, Superior de Coatzacoalcos, Colima, Cautla, Durango, Superior de El Dorado, El Llano de Aguascalientes, Huejutla, Huatabampo, Superior de Huixquilucan, Iguala, Superior de Irapuato, La Laguna, La Paz, León, Linares, Superior de Macuspana, Matamoros, Mazatlán, Mérida, Mexicali, Nuevo Laredo, Superior del Oriente del Estado de Hidalgo, Orizaba, Pachuca, Superior de Pátzcuaro, Superior de Poza	Reunión Nacional de Implementación Curricular y Fortalecimiento Curricular de las asignaturas comunes por área de conocimiento para los planes de estudio actualizados del SNEST.

Lugar y fecha de elaboración o revisión	Participantes	Evento
	Rica, Superior de Progreso, Puebla, Superior de Puerto Vallarta, Querétaro, Reynosa, Roque, Salina Cruz, Saltillo, San Luis Potosí, Superior de Tacámbaro, Superior de Tamazula de Gordiano, Tehuacán, Tijuana Tlaxiaco, Toluca, Torreón, Tuxtepec, Superior de Venustiano Carranza, Veracruz, Villahermosa, Zacatecas, Superior de Zongólica.	
Instituto Tecnológico de Aguascalientes, del 15 al 18 de Junio de 2010.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Cd. Madero, Colima, La Paz, Toluca y Villahermosa.	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.

5.- OBJETIVO GENERAL DEL CURSO

Utilizar técnicas de modelado para la solución de problemas.

Aplicar la sintaxis de un lenguaje orientado a objetos.

Aplicar un lenguaje orientado a objetos para la solución de problemas.

6.- COMPETENCIAS PREVIAS

- Analizar y solucionar problemas informáticos y representar su solución mediante herramientas de software orientado a objetos.

7.- TEMARIO

Unidad	Temas	Subtemas
1.	Fundamentos del lenguaje	1.1. Entorno de desarrollo. 1.2. Configuración del entorno de desarrollo. 1.3. Palabras reservadas. 1.4. Comentarios. 1.5. Tipos de datos. 1.6. Variables. 1.7. Constantes. 1.8. Operadores. 1.9. Sentencias. 1.10. Conversión de tipos de datos (cast). 1.11. Estructuras de control.
2.	Arreglos	2.1. Unidimensional. 2.2. Multidimensional.
3.	Clases y objetos	3.1. Definición de una clase. 3.2. Declaración de clases. 3.3. Miembros de una clase. 3.4. Ámbito referente a una clase. 3.5. Especificadores de acceso. 3.6. Creación de objetos. 3.7. Puntero this. 3.8. Constructores y destructores. 3.9. Clases Predefinidas. 3.10. Definición, creación y reutilización de paquetes/librerías. 3.11. Manejo de excepciones.
4.	Métodos	4.1. Definición de un método.

		<p>4.2. Estructura de un método.</p> <p>4.3. Valor de retorno.</p> <p>4.4. Declaración de un método.</p> <p>4.5. Ámbito y tiempo de vida de variables.</p> <p>4.6. Argumentos y paso de parámetros.</p> <p>4.7. Sobrecarga de métodos.</p> <p>4.8. Encapsulamiento.</p>
5.	Herencia y polimorfismo	<p>5.1. Concepto de herencia y polimorfismo.</p> <p>5.2. Definición de una clase base.</p> <p>5.3. Definición de una clase derivada.</p> <p>5.4. Clases abstractas.</p> <p>5.5. Definición de herencia múltiple.</p> <p>5.6. Implementación de herencia múltiple.</p> <p>5.7. Reutilización de la definición de paquetes / librerías.</p> <p>5.8. Clases genéricas (Plantillas).</p>
6.	Archivos	<p>6.1. Definición de Archivos de texto y archivos binarios.</p> <p>6.2. Operaciones básicas en archivos texto y binario.</p> <p>6.3. Manejo de excepciones en archivos.</p>

8.- SUGERENCIAS DIDÁCTICAS

El docente debe:

Ser conocedor de la disciplina que está bajo su responsabilidad, conocer su origen y desarrollo histórico para considerar este conocimiento al abordar los temas. Desarrollar la capacidad para coordinar y trabajar en equipo; orientar el trabajo del estudiante y potenciar en él la autonomía, el trabajo cooperativo y la toma de decisiones. Mostrar flexibilidad en el seguimiento del proceso formativo y propiciar la interacción entre los estudiantes. Tomar en cuenta el conocimiento de los estudiantes como punto de partida y como obstáculo para la construcción de nuevos conocimientos.

- Propiciar actividades de metacognición. Ante la ejecución de una actividad, señalar o identificar el tipo de proceso intelectual que se realizó: una identificación de patrones, un análisis, una síntesis, la creación de un heurístico, etc. Al principio lo hará el profesor, luego será el estudiante quien lo identifique.
- Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes.
- Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes. Ejemplo: Realizar practicas en equipo que permitan obtener un resultado a partir del trabajo de todos.
- Relacionar los contenidos de esta asignatura con las demás del plan de estudios a las que ésta da soporte para desarrollar una visión interdisciplinaria en el estudiante. Ejemplos: Resolver un grupo de ecuaciones por el método de Cramer.
- Propiciar el desarrollo de capacidades intelectuales relacionadas con la lectura, la escritura y la expresión oral. Ejemplos: trabajar las actividades prácticas a través de guías escritas, redactar reportes e informes de las actividades de experimentación, exponer al grupo las conclusiones obtenidas durante las observaciones.
- Facilitar el contacto directo con problemas de su entorno para que plantee la solución mediante el modelado orientado a objetos y programe la solución utilizando el lenguaje de programación orientado a objetos.
- Propiciar el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, que encaminen hacia una posición crítica del estudiante.
- Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura.
- Proponer problemas que permitan al estudiante la integración de contenidos de la asignatura y entre distintas asignaturas, para su análisis y solución.
- Relacionar los contenidos de la asignatura con el cuidado del medio ambiente; así como con las prácticas.
- Cuando los temas lo requieran, utilizar medios audiovisuales para una mejor comprensión del estudiante.
- Propiciar el uso de comando por consola en las etapas de construcción de la solución.
- Propiciar el uso de ambientes GUI.

9.- SUGERENCIAS DE EVALUACIÓN

La evaluación debe ser continua y formativa por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Reportes escritos de las observaciones hechas durante las actividades realizadas en el laboratorio, así como de las conclusiones obtenidas de dichas observaciones.
- Reportes escritos de las soluciones a problemas desarrollados fuera de clase.
- Información obtenida durante las investigaciones solicitadas plasmada en documentos escritos.
- Exámenes escritos para comprobar el manejo de aspectos teóricos.
- Desarrollo de programas de ejemplo para el uso de lenguaje orientado a objetos.
- Elaboración de proyectos donde el estudiante resuelva problemas de su entorno mediante la programación orientada a objetos.
- Descripción de otras experiencias concretas que podrían realizarse adicionalmente (participación, integración, entrega de proyectos en tiempo, etc.).

10.- UNIDADES DE APRENDIZAJE

Unidad 1: Fundamentos del lenguaje

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
Utilizar técnicas de modelado para la solución de problemas. Aplicar la sintaxis de un lenguaje orientado a objetos. Aplicar un lenguaje orientado a objetos para la solución de problemas.	<ul style="list-style-type: none">• Seleccionar e instalar el compilador.• Utilizar comandos en ambiente de consola para la compilación y ejecución de los programas.• Seleccionar e instalar un ambiente de interfaz gráfico (GUI).• Utilizar los comentarios como documentación del programa.• Realizar ejercicios donde distinga identificadores validos y no validos.• Investigue las palabras reservadas del lenguaje.• Identifique y utilice los tipos de datos básicos del lenguaje.• Realizar ejercicios donde defina literales numéricas y de texto.• Realizar ejercicios en donde identifique una variable de un tipo básico y una variable de referencia.• Realizar ejercicios de inicialización de variables.• Investigue los operadores del lenguaje.• Realizar ejercicios de compatibilidad de tipos de datos y la utilización de cast.• Realizar ejercicios utilizando las estructuras if, if-else y switch.• Realizar ejercicios utilizando las estructuras for, while y do.• Investigue las sentencias continue y break

	como estructuras de control de flujo.
--	---------------------------------------

Unidad 2: Arreglos

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
<p>Utilizar técnicas de modelado para la solución de problemas.</p> <p>Aplicar la sintaxis de un lenguaje orientado a objetos.</p> <p>Aplicar un lenguaje orientado a objetos para la solución de problemas.</p>	<ul style="list-style-type: none"> • Realizar ejercicios en donde se declaren arreglos de tipos de datos básicos. • Realizar ejercicios en donde se declaren arreglos de clases. • Realizar ejercicios de inicialización de un arreglo y la determinación del número de elementos del arreglo. • Investigue los arreglos multidimensionales.

Unidad 3: Clases y objetos

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
<p>Utilizar técnicas de modelado para la solución de problemas.</p> <p>Aplicar la sintaxis de un lenguaje orientado a objetos.</p> <p>Aplicar un lenguaje orientado a objetos para la solución de problemas.</p>	<ul style="list-style-type: none"> • Realizar ejercicios para la creación de clases y objetos. • Investigue la inicialización de un objeto. • Utilizar en ejercicios los modificadores de acceso. • Utilizar las clases definidas en el lenguaje orientado a objetos. • Realizar ejercicios de definición de excepciones como try, catch y finally.

Unidad 4: Métodos

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
<p>Utilizar técnicas de modelado para la solución de problemas.</p> <p>Aplicar la sintaxis de un lenguaje orientado a objetos.</p> <p>Aplicar un lenguaje orientado a objetos para la solución de problemas.</p>	<ul style="list-style-type: none"> • Desarrollar ejercicios en donde se definan los comportamientos de los objetos.

Unidad 5: Herencia y polimorfismo

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
<p>Utilizar técnicas de modelado para la solución de problemas.</p> <p>Aplicar la sintaxis de un lenguaje orientado a objetos.</p>	<ul style="list-style-type: none"> • Consensar en una lluvia de ideas el concepto de herencia y discutir en casos se puede aplicar. • Desarrollar y analizar las semejanzas y diferencias entre herencia y polimorfismo

<p>Aplicar un lenguaje orientado a objetos para la solución de problemas.</p>	<p>para determinar las posibles relaciones entre ellos.</p> <ul style="list-style-type: none"> • Crear aplicaciones en donde se pueda demostrar la sobrecarga de métodos y razonar su utilidad. • Aplicar la reutilización de código para el diseño de clases nuevas a partir de clases ya probadas. • Estudiar y comprender el concepto de interfaces para implementar la herencia múltiple en el diseño de clases. • Crear clases y jerarquías de clases para aplicarlas a problemas reales. • Plantear y resolver problemas informáticos mediante la utilización del polimorfismo.
---	--

Unidad 6: Archivos

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
<p>Utilizar técnicas de modelado para la solución de problemas.</p> <p>Aplicar la sintaxis de un lenguaje orientado a objetos.</p> <p>Aplicar un lenguaje orientado a objetos para la solución de problemas.</p>	<ul style="list-style-type: none"> • Investigar en fuentes de información los conceptos y metodologías para manipular archivos de texto y binarios en un lenguaje de programación orientado a objetos. • Programar una clase que cree, consulte, modifique y borre archivos de texto. • Programar una clase que cree, consulte, modifique y borre archivos binarios. • Diseñar un caso de estudio que requiera el uso de archivos para que sea resuelto por el alumno. • Desarrollar ejercicios que definan archivos de entrada y salida.

11.- FUENTES DE INFORMACIÓN

1. Joyanes Aguilar, Luis. Java 2: Manual de Programación. 4ª edición. Ed. Prentice Hall.
2. Joyanes Aguilar, Luis. Programación en Java 2 Algoritmos, Estructura de Datos y Programación Orientada a Objetos. Ed. Prentice Hall.
3. Eckel, Bruce. Thinking in Java. 3a edición. Ed. Prentice Hall.
4. Martin, Robert C. UML para Programadores Java. Ed. Pearson Educación.
5. Pilone, Dan y Pitman, Neil. UML 2.0 in a Nutshell. Ed. O'Reilly.
6. Sierra, Katherine. [SCJP Sun Certified Programmer for Java 6](#). Ed. McGraw Hill.

12.- PRÁCTICAS PROPUESTAS

Es recomendable la realización de prácticas en todas las unidades que consistan en el modelado y resolución de problemas utilizando un lenguaje de programación orientado a objetos; la entrega al final de cada unidad de un proyecto que refuerce en la aplicación de los temas vistos en clase y la entrega de un proyecto final que converja en la aplicación de los conceptos vistos en la materia, el cual debe ser definido al final de la primera unidad.

- Realizar programas: conversión de unidades métricas, monedas, tablas de multiplicar, en función a una calificación imprimir un mensaje, etc.
- Realizar programas: ordenamiento de números en un arreglo unidimensional o bidimensional utilizando el método de burbuja, resolver la determinante de una matriz, etc.
- Realizar programas en donde construyan las clases de problemas dados y relacionados con otras materias, 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 materias, 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 que la entrada y salida de los datos se almacenen en archivos.