

**PROGRAMA DE ESTUDIOS POR COMPETENCIAS
PROGRAMACIÓN ORIENTADA A OBJETOS**

I. IDENTIFICACIÓN DEL CURSO

Espacio Educativo: Facultad de Ingeniería						
Licenciatura: Ingeniería de Computación				Área de docencia: Programación de Ingeniería de Software		
Año de aprobación por el Consejo Universitario:						
Aprobación por los HH. Consejos Académico y de Gobierno		Fecha:		Programa elaborado por: Ing. Bertha Ordoñez Guadarrama Ing. Francisco Ignacio Chávez Castañeda		Programa revisado por:
				Fecha de elaboración : Octubre 2013		
Clave	Horas de teoría	Horas de práctica	Total de horas	Créditos	Tipo de curso	Núcleo de formación
L41007	3	2	5	8	Curso-Laboratorio	Sustantivo
Unidad de Aprendizaje Antecedente Programación Estructurada, Estructuras de datos, Organización de Archivos				Unidad de Aprendizaje Consecuente Teoría de Sistemas		
Programas educativos o espacios académicos en los que se imparte: Licenciatura en Ingeniería en Computación (Facultad. de Ingeniería, Centros Universitarios: Atlacomulco, Ecatepec, Texcoco, Valle de Chalco, Valle de México, Valle de Teotihuacán, Zumpango)						

II. PRESENTACIÓN DEL PROGRAMA

El conocimiento en el desarrollo de aplicaciones basadas en el paradigma de programación orientado a objetos, se vuelve un requisito en la formación académica del ingeniero en sus diferentes ramas. Con el manejo de un lenguaje de programación orientada a objetos, el estudiante podrá implementar los conceptos de orientación a objetos y desarrollar, en cursos consecuentes, un número significativo de aplicaciones, en las diferentes plataformas que así lo soliciten.

Por lo previamente expuesto y como parte de una formación integral para el ingeniero en computación, se propone el siguiente programa que consta de cinco unidades de competencia, en la primera se consideran los fundamentos teóricos sobre un lenguaje de POO y la plataforma en la que ejecuta, dentro de la segunda se abordan conocimientos sobre la sintaxis básica del lenguaje de POO y como parte de la tercera se refiere básicamente a controlar el flujo del programa.

En la unidad cuatro se crearan y manipularán estructuras dinámicas de datos utilizando referencias, se conocerán los procedimientos para el manejo de archivos de acceso secuencial y acceso aleatorio. En la cinco se entenderán los principios de diseño de las interfaces gráficas de usuario (GUI), clases e interfaces para manejo de eventos y propiedades, tomando como ejemplo práctico el manejo de archivos.

Para lograr lo anterior se sugieren como estrategias didácticas la revisión bibliográfica y solución de ejercicios por parte del alumno y la explicación por parte del instructor de temas específicos de mayor complejidad. Para consolidar los conocimientos, también es necesario realizar ejercicios que fortalezcan la parte teórica.

La evaluación de la unidad de aprendizaje se efectuará a través de cinco exámenes parciales escritos, así mismo se considerará la puntual asistencia a clase y la presentación de un proyecto final desarrollado por el alumno en el cual se apliquen los conocimientos adquiridos durante el curso.

Con lo antes mencionado, el ingeniero en computación como experto en su ramo tendrá las herramientas necesarias para poder interactuar de manera holística con profesionales en otros campos del saber para así solucionar problemas en bases científico-metodológicas congruentes afrontando los retos actuales del desarrollo tecnológico y económico.

III. LINEAMIENTOS DE LA UNIDAD DE APRENDIZAJE

DEL DOCENTE	DEL DISCENTE
<ul style="list-style-type: none">- Realizar el encuadre del curso.- Asistir puntualmente a las clases.- Preparar material didáctico para las clases.- Asesorar a los alumnos cuando sea necesario.- Resolver las dudas de los alumnos.- Evaluar la unidad de aprendizaje.- Entregar resultados de las evaluaciones.- Participar en la realización de exámenes departamentales.- Entregar en tiempo y forma el resultado de evaluaciones parciales, ordinaria, extraordinaria y a título de suficiencia.	<ul style="list-style-type: none">- Asistir puntualmente a clases.- Contar con el 80% de asistencia para presentar examen ordinario- Contar con el 60% de asistencia para presentar examen extraordinario- Contar con el 30% de asistencia para presentar examen a título de suficiencia- Asumir una actitud participativa en las sesiones de clase.- Realizar las evaluaciones que se establezcan.- Participar activamente y entregar en tiempo y forma los trabajos extraclase- Tener sentido de responsabilidad en los trabajos extraclase- Entregar en tiempo y forma los trabajos extraclase- Tener sentido de integración y participación dentro del salón de clases.

IV. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

El alumno: Conocerá los conceptos de la programación orientado a objetos y su implementación en un lenguaje apropiado, los cuales servirán de base para unidades de aprendizaje encaminadas al análisis, el diseño y la elaboración de aplicaciones informáticas.

V. COMPETENCIAS GENÉRICAS

- Contar con bases para comprometerse con la calidad.
- Lograr una comunicación oral y escrita eficaz en su propia lengua y en una segunda lengua.
- Utilizar eficazmente al menos un lenguaje de programación orientado a objetos.
- Responder eficazmente a nuevas situaciones informáticas.
- Comunicarse con expertos de otras áreas.
- Aplicar los conocimientos en la práctica.
- Crear nuevas ideas para la solución de problemas.
- Dar mantenimiento a software.
- Evaluar sistemas.
- Desarrollar software.

VI. ÁMBITOS DE DESEMPEÑO PROFESIONAL

- Empresas públicas y privadas dedicadas al desarrollo de aplicaciones.
- Diseño y desarrollo de proyectos software.
- Implementación y mantenimiento de sistemas computacionales orientados a objetos.
- Sector educativo.

VII. ESCENARIOS DE APRENDIZAJE

- Salón de Clases.
- Sala de Cómputo.

VIII. ESTRUCTURA DE LA UNIDAD DE APRENDIZAJE

1. Comprender los aspectos históricos y tecnológicos que dan importancia al desarrollo de software orientado a objetos.
2. Implementar los conceptos básicos de la POO (encapsulamiento, herencia y polimorfismo).
3. Implementar los conceptos avanzados de la POO.
4. Utilizar un lenguaje de modelado estándar para lograr una documentación apropiada del software.
5. Instrumentar sistemas de software que mediante la aplicación de POO.

IX. DESARROLLO DE LA UNIDAD DE APRENDIZAJE

UNIDAD DE COMPETENCIA I	ELEMENTOS DE COMPETENCIA		
	Conocimientos	Habilidades	Actitudes/ Valores
Comprender los aspectos históricos y tecnológicos que dan importancia al desarrollo de software orientado a objetos.	<p>Introducción</p> <ul style="list-style-type: none"> - Crisis del software - Criterios de calidad del software - Evolución de la programación, paradigma y metodología <p>Conceptos básicos de la P.O.O.</p> <ul style="list-style-type: none"> - Reutilización - Robustez - Clases y Objetos - Polimorfismo de sobrecarga - Herencia - Polimorfismo de sobre-escritura <p>Lenguajes Orientados a Objetos</p> <ul style="list-style-type: none"> - Características generales de los LPOO. - Historia de los LPOO - Comparación de los más populares LPOO - El lenguaje Java 	<p>Observación</p> <p>Conceptualización</p> <p>Análisis</p>	<ul style="list-style-type: none"> - Responsabilidad - Actitud prepositiva, constructivista e innovadora - Tolerancia y participación Trabajo en equipo Receptiva Analítica

	- Compilación y ejecución de programas en Java. - Documentación de la API		
Estrategias didácticas: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Discusión en grupo	Recursos requeridos Pizarrón Libro de texto Computadora Videoprojector Diapositivas Videos	Tiempo destinado 5 hrs.	
CRITERIOS DE DESEMPEÑO I	EVIDENCIAS		
	DESEMPEÑO	PRODUCTOS	
- Dominio de conceptos básicos del paradigma de orientación a objetos - Comprensión de la Crisis del Software - Evaluación con base en los criterios de calidad del software - Síntesis de las características de los LPOO		Ensayos Programas	

UNIDAD DE COMPETENCIA II	ELEMENTOS DE COMPETENCIA		
	Conocimientos	Habilidades	Actitudes/ Valores
Implementar los conceptos básicos de la POO (encapsulamiento, herencia y polimorfismo).	Clases y Objetos <ul style="list-style-type: none"> - Objetos - Mensajes - Clases - Encapsulamiento. - Atributos y métodos. - Variables de referencia. - Constructoras. - Firma de métodos. - Polimorfismo de sobrecarga - Ocultamiento de información. - Estado concreto y abstracto de los objetos. - Identidad y copia de objetos. - Valores y funciones de clase Herencia <ul style="list-style-type: none"> - Concepto. - Implementación. - Implicaciones constructoras. - El uso de super. - Clase Object - Polimorfismo de sobre-escritura 	Programación de aplicaciones, usando un lenguaje de programación orientado a objetos.	<ul style="list-style-type: none"> - Responsabilidad - Actitud prepositiva, constructivista, receptiva, analítica e innovadora - Tolerancia y participación - Trabajo en equipo
Estrategias didácticas: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Discusión en grupo Solución de ejercicios Desarrollo de programas	Recursos requeridos Pizarrón Libro de texto Computadora Videoprojector Diapositivas Videos	Tiempo destinado 20 hrs.	

CRITERIOS DE DESEMPEÑO II	EVIDENCIAS	
	DESEMPEÑO	PRODUCTOS
	Conocimiento de los pilares sobre los que se basa el paradigma de O.O.	Ejercicios y tareas
Aplicación del encapsulamiento, herencia y polimorfismo	Manejo de un lenguaje de programación O.O.	Desarrollo de práctica y reporte de pequeñas aplicaciones bajo el paradigma O.O

UNIDAD DE COMPETENCIA III	ELEMENTOS DE COMPETENCIA		
	Conocimientos	Habilidades	Actitudes/ Valores
Implementar los conceptos avanzados de la POO.	<p>Relaciones entre clases</p> <ul style="list-style-type: none"> - Asociación. - Agregación. - Composición. - Generalización. - Dependencia. <p>Manejo de excepciones y asserts</p> <ul style="list-style-type: none"> - Definición - Manejo - Declaración de nuevos tipos de excepciones. - Aserciones (Assets). <p>Interfaces</p> <ul style="list-style-type: none"> - Clases abstractas. - Interfaces. - Implementación de interfaces. - Representación UML <p>Herencia</p> <ul style="list-style-type: none"> - Herencia múltiple - El problema de identificadores duplicados. - El problema del ancestro común. - Herencia selectiva. - Herencia virtual - Variables y métodos final <p>Genericidad</p> <ul style="list-style-type: none"> - Clases genéricas. - Declaración de clases genéricas. - Uso de clases genéricas. 	Programación de aplicaciones, usando un lenguaje de programación orientado a objetos.	<ul style="list-style-type: none"> - Responsabilidad - Actitud prepositiva, constructivista, receptiva, analítica e innovadora - Tolerancia y participación - Trabajo en equipo

	Persistencia Colecciones e Iteradores		
Estrategias didácticas: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Discusión en grupo Solución de ejercicios Desarrollo de programas		Recursos requeridos Pizarrón Libro de texto Computadora Videoprojector Diapositivas Videos	Tiempo destinado 35 hrs.
CRITERIOS DE DESEMPEÑO III	EVIDENCIAS		
	DESEMPEÑO	PRODUCTOS	
Conocimiento de los pilares sobre los que se basa el paradigma de O.O.	Ejercicios y tareas	Identificación de conceptos y principios del paradigma Orientado a Objetos	
Aplicación de los conocimientos de la Unidad III	Manejo de un lenguaje de programación O.O.	Desarrollo de práctica y reporte de pequeñas aplicaciones bajo el paradigma O.O	

UNIDAD DE COMPETENCIA IV	ELEMENTOS DE COMPETENCIA		
	Conocimientos	Habilidades	Actitudes/ Valores
Utilizar un lenguaje de modelado estándar para lograr una documentación apropiada del software.	Importancia de la documentación Documentación Técnica - Diagrama de casos de uso - Diagrama de clases - Diagrama de secuencia Manual de usuario	Conceptualización Análisis Redacción y presentación de documento	- Responsabilidad - Actitud prepositiva, constructivista, receptiva, analítica e innovadora - Tolerancia y participación - Trabajo en equipo
Estrategias didácticas: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Discusión en grupo Solución de ejercicios Desarrollo de programas	Recursos requeridos Pizarrón Libro de texto Computadora Videoprojector Diapositivas Videos Software especializado (umbrello, racional rose, dia, ArgoUML)	Tiempo destinado 10 hrs.	
CRITERIOS DE DESEMPEÑO IV	EVIDENCIAS		
		DESEMPEÑO	PRODUCTOS
Aplicación de los conocimientos de la unidad	Uso de los diagramas UML	Documentación de un proyecto	

UNIDAD DE COMPETENCIA V	ELEMENTOS DE COMPETENCIA		
	Conocimientos	Habilidades	Actitudes/ Valores
Instrumentar sistemas de software mediante la aplicación de POO.	<ul style="list-style-type: none"> - MVC/PAC - Manejo de eventos - Uso de objetos para la creación GUIs 	<ul style="list-style-type: none"> Conceptualización Análisis Creatividad 	<ul style="list-style-type: none"> - Responsabilidad - Actitud prepositiva, constructivista, receptiva, analítica e innovadora - Tolerancia y participación - Trabajo en equipo
Estrategias didácticas: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Discusión en grupo Solución de ejercicios Desarrollo de programas		Recursos requeridos Pizarrón Libro de texto Computadora Videoprojector Diapositivas Videos IDE (Eclipse, Netbeans o equivalente)	Tiempo destinado <p style="text-align: center;">10 hrs.</p>
CRITERIOS DE DESEMPEÑO V	EVIDENCIAS		
		DESEMPEÑO	PRODUCTOS
Aplicación de los conocimientos de la unidad		Manejo de un lenguaje de programación O.O. y de un IDE	Sistema de software

X. EVALUACIÓN Y ACREDITACIÓN

Cumplir con el reglamento que respecto a asistencias, establece la misma Universidad. (Del total de clases 80% para derecho a ordinario y 60% para extraordinario)

EVALUACIÓN:

Para obtener la calificación se considerará:

Exámenes parciales escritos (se considera la posibilidad de exámenes rápidos o sorpresa)

Tareas de investigación

Proyecto de aplicación

Si el promedio obtenido mediante estos instrumentos es mayor o igual a 8.0 el alumno se considerará exento de presentar evaluación ordinaria.

Evaluación ordinaria:

Exámenes 70 % Proyecto final 30 %

Evaluación extraordinaria:

Examen acumulativo 50 % Proyecto final 50 %

Evaluación a título de suficiencia

Examen acumulativo 50 % Proyecto final 50 %

La evaluación del proyecto final califica los siguientes aspectos:

1. Estilo de programación.
2. Encapsulamiento y ocultamiento de información.
3. Manejo de excepciones.
4. Genericidad.
5. Interfaces.
6. Herencia y agregación.
7. Variables y funciones de clase
8. Persistencia.
9. Funcionamiento
10. Documentación.
11. Aplicación de UML

XI. REFERENCIAS

BIBLIOGRAFÍA BÁSICA

- Wu Thomas. (2008) Introducción a la programación orientada a objetos. Mc Graw Hill.
- Deytel Harvey M y Paul J. Deytel. (2008) Como programar en Java, 7ª edición, Prentice Hall México.
- Dózek Adam. (2007) Estructuras de datos y algoritmos con Java. 2ª edición. Thomson Learning.
- Goodrich Michel T. y Tamassia Roberto. (2002) Estructuras y datos y algoritmos en Java, 2ª edición, CECSA, México.
- Staugaard Jr. Andrew C. (1998) Técnicas estructuradas y orientadas a objetos, Prentice Hall, México.
- López R. Leobardo. (2006) Metodología de la programación Orientada a Objetos. Alfaomega.

BIBLIOGRAFÍA COMPLEMENTARIA

- David Arnow, Gerald Weiss. (2001) Introducción a la programación con java, un enfoque orientado a objetos, Addison Wesley.
- Deitel y Deitel. (1999) C++ Cómo programar, segunda edición, Prentice Hall
- Joyanes Aguilar L. (1998) Programación Orientada a Objetos, segunda edición, McGraw Hill
- Jo Ann Smith. (2001) C++ Desarrollo de proyectos con programación orientada a objetos, Thomson Learning, México
- Booch, Jacobson, Rumbaugh. (1999) El lenguaje unificado de Modelado, Addison Wesley
- Booch G. Rumbaugh J. y Jacobson I. (2006) El lenguaje unificado de modelado. Pearson Educación. Madrid.
- Jacobson, Booch, Rumbaugh. (2000) El proceso Unificado de Desarrollo de Software, Addison Wesley.
- Herbert Schildt. (2005) Biblia de Java 5, Anaya