

DATOS DE IDENTIFICACIÓN

MATERIA:	Programación Orientada a Objetos				
CENTRO ACADÉMICO:	Ciencias Básicas				
DEPARTAMENTO ACADÉMICO:	Sistemas de Información				
PROGRAMA EDUCATIVO:	Licenciatura en Informática y Tecnologías Computacionales				
AÑO DEL PLAN DE ESTUDIOS:	2014	SEMESTRE:	Cuarto	CLAVE DE LA MATERIA:	22355
ÁREA ACADÉMICA:	Computación Básica	PERIODO EN QUE SE IMPARTE:	Enero-Junio		
HORAS SEMANA T/P:	3/2	CRÉDITOS:	8		
MODALIDAD EDUCATIVA EN LA QUE SE IMPARTE:	Presencial	NATURALEZA DE LA MATERIA:	Teórico-Práctica		
ELABORADO POR:	Dr. Juan Muñoz López				
REVISADO Y APROBADO POR LA ACADEMIA DE:	Computación Básica	FECHA DE ACTUALIZACIÓN:	Enero 2017		

DESCRIPCIÓN GENERAL

Curso teórico-práctico de modalidad presencial centrado en la solución de problemas mediante el manejo de un lenguaje de Programación Orientado a Objetos (POO). La importancia de este curso radica principalmente en conocer y desarrollar aplicaciones para la resolución de problemas en el campo laboral mediante sistemas de información utilizando como base el paradigma de la POO. Esta materia tiene como precedentes a Algoritmos Computacionales, Programación Estructurada, Programación Orientada a Eventos, Estructura de Datos y es antecedente de Base de Datos e Ingeniería de Usabilidad y Diseño de Software, Análisis y Diseño de Sistemas e Ingeniería de Software.

OBJETIVO (S) GENERAL (ES)

Al finalizar el curso, el estudiante comprenderá los conceptos de la POO y su implementación por medio de un lenguaje, enfatizando la importancia de la creación, reutilización y optimización de aplicaciones basadas en el uso de clases para la creación de aplicaciones modulares y flexibles que permitan reducir costos y tiempos de desarrollo, realizándolo de manera innovadora, creativa y con calidad.

CONTENIDOS DE APRENDIZAJE

UNIDAD TEMÁTICA I: Fundamentos de Java (horas aprox.)		
OBJETIVOS PARTICULARES	CONTENIDOS	FUENTES DE CONSULTA
El alumno será capaz de: • Describir las características de un lenguaje bajo el paradigma de Programación	1. Introducción a Java 1.1. Antecedentes de Java 1.2. La Máquina Virtual de Java y el bytecode 1.2.1. Compilación y ejecución de programas 1.2.2. Entornos de desarrollo de Java 1.2.3. Tecnologías de Java	[2] [4] [6] [7]



<p>Orientada a Objetos (POO)</p> <ul style="list-style-type: none"> Comprender la organización de un Programa bajo el paradigma de POO Comprender la organización de un programa Orientado a Objetos Describir los Métodos y estructuras de control en Java Comprender el manejo de arreglos y cadenas en POO Comprender las consideraciones para el manejo de excepciones y la validación 	<ol style="list-style-type: none"> Programación Orientada a Objetos en Java <ol style="list-style-type: none"> Tipos de aplicaciones en Java Principios de programación en Java Abstracción Clases y objetos Métodos Comentarios Manejo de basura Control de acceso Elementos del Lenguaje de Programación Java <ol style="list-style-type: none"> Palabras reservadas Tipos de datos, literales y constantes Declaración, inicialización y uso de variables <ol style="list-style-type: none"> Ámbito Alcance y visibilidad Manejo de la clase Scanner Operadores y asignaciones Control de flujo Manejo de arreglos Manejo de cadenas Manejo de excepciones Validación Afirmaciones 	<p>[11] [15]</p>
---	---	----------------------

UNIDAD TEMÁTICA II: Aplicación de los Conceptos de la Programación Orientada a Objetos (horas aprox.)		
OBJETIVOS PARTICULARES	CONTENIDOS	FUENTES DE CONSULTA
<p>El alumno será capaz de:</p> <ul style="list-style-type: none"> Comprender los fundamentos del paradigma de la Programación Orientada a Objetos Aplicar los fundamentos del paradigma de la Programación Orientada a Objetos a programas desarrollados en Java Comprender y aplicar las consideraciones para el diseño y desarrollo de clases 	<ol style="list-style-type: none"> Estructura de un Programa Orientado a Objetos en Java <ol style="list-style-type: none"> Forma general de una clase <ol style="list-style-type: none"> Declaración e instanciación Valores de referencia y referencias Miembros de una clase <ol style="list-style-type: none"> Miembros de instancia Miembros de clase (estáticos) Atributos y Métodos <ol style="list-style-type: none"> Concepto y Declaración de Atributos Concepto y Declaración de Métodos Sintaxis Paso de mensajes Tipos de Métodos Sobrecarga Estándares de codificación Fundamentos de la Programación Orientada a Objetos <ol style="list-style-type: none"> Encapsulamiento Relaciones entre clases (Dependencia, Asociación, Composición, Agregación, Herencia) Herencia y agregación <ol style="list-style-type: none"> Jerarquía de Clases Clases y Métodos Abstractos Redefinición (sobre-escritura) Manejo de Constructores Tipos genéricos y manejo de comodines Clases abstractas 	<p>[1] [2] [3] [4] [6] [11] [12] [13] [14] [15] [16]</p>



	3.5. Polimorfismo 3.5.1. Interfaces	
--	--	--

UNIDAD TEMÁTICA III Reutilización de Código y Persistencia de Datos (horas aprox.)		
OBJETIVOS PARTICULARES	CONTENIDOS	FUENTES DE CONSULTA
El alumno será capaz de: <ul style="list-style-type: none"> Comprender el concepto de reutilización Aplicar el concepto de reutilización mediante la construcción de paquetes en Java Comprender los mecanismos de entrada y salida en Java y su relación con la persistencia de datos Describir la tecnología de acceso a datos persistentes en Java 	1. Concepto de reutilización 2. Paquetes <ul style="list-style-type: none"> 2.1. Creación 2.2. Paquetes de uso general 2.3. Clases de envoltorio 3. Concepto de persistencia de datos 4. Manejo de Entrada/Salida <ul style="list-style-type: none"> 4.1. Manejo de Archivos y Directorios 4.2. Flujos de Entrada/salida 4.3. Seriación 5. Acceso a datos <ul style="list-style-type: none"> 5.1. Tecnología JDBC 5.2. Conexión a bases de datos 5.3. Acceso a datos con JDBC 	[1] [2] [5] [6] [7] [11] [15]

METODOLOGÍA DE ENSEÑANZA - APRENDIZAJE

El curso se desarrolla mediante exposiciones por parte del maestro, contando con la participación del alumno para la discusión y análisis de conceptos.

Se incluye una parte práctica que será dirigida por el maestro en la que el alumno deberá desarrollar las aplicaciones que le permitan ejemplificar los conceptos expuestos.

Para complementar el curso, el alumno deberá realizar sesiones de autoestudio, tareas, trabajos de investigación y proyectos de aplicación que le permitirán reforzar y ampliar los conocimientos obtenidos en clase.

RECURSOS DIDÁCTICOS

- Exposición verbal por parte del maestro
- Desarrollo de ejemplos por parte del maestro
- Desarrollo de programas de software por parte del alumno en el que se aplicarán los conceptos aprendidos en clase complementados por investigación
- Realización de investigaciones por parte del alumno
- Autoestudio de conceptos dirigido por el maestro
- Elaboración de proyectos de aplicación práctica por parte del alumno
- Asesorías por parte del maestro

EVALUACIÓN DE LOS APRENDIZAJES



La parte teórica representa el 60% de la calificación final y se divide en los siguientes rubros:

- 2 Exámenes parciales (incluyen material de clases y de autoestudio) 20% c/u
- 1 Examen final global (incluye todo el material de clases y de autoestudio) 40%
- Ejercicios de investigación 20%

La parte práctica representa el 40% de la calificación final y se divide en los siguientes rubros:

- Tareas 50%
- Revisión de avances de proyecto integrador (entrega de medio término) 20%
- Revisión final del proyecto integrador (entrega final) 30%

Es requisito del curso aprobar por separado tanto la parte teórica como la parte práctica. En caso de no aprobar alguna de las dos partes se asentará como calificación final la calificación obtenida en la parte no aprobada.

No se aplicarán exámenes fuera de las fechas programadas y acordadas con el grupo, salvo causas justificadas.

Para obtener derecho a examen final, el alumno deberá asistir al menos al 80% de las sesiones programadas.

Se recomienda hacer al principio del semestre un examen diagnóstico que cubra conceptos de Algoritmos Computacionales, Programación Estructurada, Programación Orientada a Eventos y Estructura de Datos; de los resultados se harán recomendaciones individualizadas al alumno sobre aspectos a fortalecer.

FUENTES DE CONSULTA

BÁSICAS:

- [1] Antonio J. Martín Sierra; Programador Certificado Java 2. Curso Práctico. 2ª Edición; Alfaomega/Ra-Ma; 2008; México
- [2] Bart Baesens, Aimee Backiel, Seppen vanden Broucke; **Beginning Java Programming. The Object-Oriented Approach**; 2015, USA
- [3] C. Thomas Wu; **An Introduction to Object-Oriented Programming with Java. Fifth Edition**; Mc Graw Hill; 2010; USA
- [4] Fco. J. Ceballos; **Java 2, Curso de Programación, 4ta Edición**; Ra-Ma; 2010; España
- [5] Fco. Javier Ceballos; **Java 2, Interfaces Gráficas y Aplicaciones para Internet**; Alfaomega/Ra-Ma; 2005; México
- [6] P. J. Deitel & H. M. Deitel; **Java Cómo Programar, Novena edición**; Pearson Educación de México; 2012; México
- [7] Patrick Niemeyer & Daniel Leuck; **Learning Java. Fourth Edition**; O'Reilly; 2013; USA

COMPLEMENTARIAS:

- [11] Doug Lowe; **Java All-in-one for dummies. 4th Edition**; John Wiley and Sons Inc.; 2014; USA
- [12] Kathy Sierra, Bert Bates; **SCJP Sun® Certified Programmer for Java™ 6 Study Guide**; McGrawHill; 2008; USA
- [13] Mala Gupta; **OCA Java SE 7, Certification Guide**; Manning Publications; 2013; USA
- [14] Pradeep Chopra; **Java certification success, Part 1: SCJP**; IBM Co.; 2003; USA
- [15] Robert Ligouri and Patricia Ligouri; **Java 8 Pocket Guide**; O'Reilly; 2014; USA
- [16] S. G. Ganesh, Tushar Sharma; **Oracle Certified Professional; Java SE 7, Programmer Exams 1Z0-804 and 1Z0-805**; Apress; 2013; USA