



**PLAN DE ESTUDIOS (PE): Licenciatura en Ciencias de la Computación /
Ingeniería en Ciencias de la Computación**

ÁREA: Ciencias de la Computación / Ingeniería en Ciencias en Computación

ASIGNATURA: Programación II

CÓDIGO: CCOS 010

CRÉDITOS: 6

FECHA: 13 de marzo de 2017





1. DATOS GENERALES

Nivel Educativo:	Licenciatura
Nombre del Plan de Estudios:	Licenciatura en Ciencias de la Computación / Ingeniería en Ciencias de la Computación
Modalidad Académica:	Presencial
Nombre de la Asignatura:	Programación II
Ubicación:	Nivel Básico
Correlación:	
Asignaturas Precedentes:	Programación I
Asignaturas Consecuentes:	Estructuras de Datos, Bases de Datos / Estructuras de Datos, Graficación, Ingeniería de Software

2. CARGA HORARIA DEL ESTUDIANTE

Concepto	Horas por semana		Total de horas por periodo	Total de créditos por periodo
	Teoría	Práctica		
Horas teoría y práctica (16 horas = 1 crédito)	3	2	90	6

3. REVISIONES Y ACTUALIZACIONES

Autores:	Beatriz Beltrán Martínez Graciano Cruz Almanza Hilda Castillo Zacatelco Carmen Cerón Garnica José Andrés Vázquez Flores Erica Vera Cervantes Judith Pérez Marcial Laura Cuayahuitl Romero	Darnes Vilariño Ayala Marco Antonio Soriano Ulloa Rafael De la Rosa Flores Meliza Contreras González Miguel Rodríguez Hernández Pedro Bello López José Luis Meza León Yolanda Moyao Martínez
Fecha de diseño:	10 de Junio de 2009	
Fecha de la última actualización:	13 de marzo de 2017	
Fecha de aprobación por parte de la academia de área, departamento u	22 de marzo de 2017	





otro.		
Revisores:	Ana P. Cervantes Márquez Beatriz Beltrán Martínez Carmen Cerón Garnica Darnes Vilariño Ayala Erika Bonfil Barragán Eugenia Erica Vera Cervantes José Andrés Vázquez Flores Guillermina Sánchez Román Hilda Castillo Zacatelco Ma. del Carmen Santiago Díaz	Leticia Mendoza Alonso Marco Antonio Soriano Ulloa Mario Rossainz López Meliza Contreras González Miguel Rodríguez Hernández Mireya Tovar Vidal Pedro Bello López Rafael De la Rosa Flores Yalú Galicia Hernández Yolanda Moyao Martínez
Sinopsis de la revisión y/o actualización:	Se consideró la necesidad de modificar el orden de los temas en las unidades 1, 2, 3 y 4 y se eliminaron aquellos temas repetidos en otras asignaturas. También se agregó la unidad 5 relativa al tema de recursividad por la relevancia del tema; como una unidad aparte. Se agregaron y detallaron los subtemas a impartir en la unidad 4 con respecto a cada uno de los puntos, a fin de que sea más claro de qué temas impartir. Se realizó el cambio de formato y se pasó a competencias.	

4. PERFIL DESEABLE DEL PROFESOR (A) PARA IMPARTIR LA ASIGNATURA:

Disciplina profesional:	Ciencias de la Computación
Nivel académico:	Maestría
Experiencia docente:	Mínima 2 años
Experiencia profesional:	Mínima 1 año

5. PROPÓSITO:

El estudiante obtendrá los conocimientos y habilidades para resolver problemas, utilizando el paradigma orientado a objetos, mediante el uso de un lenguaje de programación, fomentando el trabajo en equipo de manera responsable, honesta, comprometido con su aprendizaje y su autonomía.

El estudiante será capaz de:

- Identificar los conceptos básicos del paradigma orientado a objetos.
- Construir abstracciones del mundo real al paradigma orientado a objetos.
- Utilizar los elementos que conforman un lenguaje de programación orientado a objetos para resolver problemas haciendo uso de un IDE.
- Aplicar los elementos avanzados de la POO para resolver un problema de la manera más adecuada.
- Utilizar la técnica de recursividad para resolver problemas computacionales





6. COMPETENCIAS PROFESIONALES:

Licenciatura

Desarrollar de forma efectiva y eficiente los algoritmos y programas apropiados para resolver problemas complejos de computación.

Aplicar fundamentos matemáticos, principios algorítmicos y teorías de Ciencias de la Computación para el modelado y diseño de soluciones computacionales.

Ingeniería

Diseñar soluciones de sistemas de cómputo soportadas en modelos de procesos, metodologías y herramientas para resolver problemas.

Diseñar soluciones creativas e innovadoras por medio del análisis, síntesis e implementación en sistemas de cómputo que cumplan con los estándares de calidad.

En la materia se modelan y desarrollan programas mediante el uso de un lenguaje de alto nivel para la resolución de problemas a través del paradigma orientado a objetos.

7. CONTENIDOS TEMÁTICOS

Unidad de Aprendizaje	Contenido Temático	Referencias
1. Conceptos básicos del paradigma Orientado a Objetos	1.1 Clase y Objeto	Joyanes, A. (2011). Programación en Java 6. Algoritmos, programación orientada a objetos e interfaz gráfica de usuario (3ª ed). Mc Graw Hill. James Martin, James J. (2007), Análisis y Diseño Orientado a Objetos (3ª. Ed.), Prentice Hall. Farrell J. (2011). Programming Logic and design comprehensive (Sixth Edition). Course Technology, Cengage Learning. Wu T. (2010). An introduction to Object-Oriented Programming with Java (Fifth Edition). Mc Graw Hill. Higher Education.
	1.2 Abstracción de datos.	
	1.3 TDA	
	1.4 Encapsulamiento	
	1.5 Herencia	
	1.6 Polimorfismo	

Unidad de Aprendizaje	Contenido Temático	Referencias
2. Abstracción del mundo	2.1 Principios básicos de modelado de objetos.	Rumbaugh, James, Blaha, Michael, Premerlani, William, Modelado y Diseño Orientado a Objetos, España: Pearson
	2.2 Modelar clases y objetos	





Unidad de Aprendizaje	Contenido Temático	Referencias
real al paradigma Orientado a Objetos	2.3 Modelar relaciones entre clases 2.3.1. Dependencia 2.3.2 Asociación 2.3.3. Agregación y composición 2.3.4. Herencia e interfaces	Educación. Larman Graig, (2004). UML y Patrones. España. Prentice Hall. Brett McLaughlin, Gary Police, (2006). Object Oriented Analysis & Design. O'Really. Primera Edición.

Unidad de Aprendizaje	Contenido Temático	Referencias
3. Lenguaje de Programación Orientado a Objetos	3.1 Tipos de datos primitivos 3.2 Declaración de clases 3.3 Relaciones entre clases 3.4 Declaración de herencia e interfaces 3.5 Polimorfismo 3.5.1. Clases Abstractas 3.5.2. Sobrecarga 3.5.3. Sobreescritura	Joyanes, L., Zahonero, I. (2011), Programación en Java 6: Algoritmos, Programación Orientada a Objetos e Interfaz Gráfica de usuario. McGraw Hill. Liang Y.D. (2014). Introduction to Java Programming Comprehensive Version. Tenth Edition. Pearson. Deitel, H. M. y Deitel, P. J. (2011), <i>How to Programming</i> (9ª. Ed.) Pearson Educación. Moldes T.J. (2011). JAVA 7. Manual Imprescindible. Editorial ANAYA. Booch G., Maksimchuk R.A., et al. (2007). Object-Oriented Analysis and Design with Applications (3rd Edition). Addison Wesley. Farrell J. (2014). Java Programming, Seventh Edition. Course Technology, Cengage Learning.

Unidad de Aprendizaje	Contenido Temático	Referencias
4. Lenguaje de Programación Orientado a Objetos: Elementos avanzados.	4.1 Tratamiento de excepciones 4.1.1 Excepciones comprobadas 4.1.2 Excepciones no comprobadas 4.1.3 Excepciones personalizadas	Ceballos F.J. (2011). Java 2. Curso de Programación. Cuarta Edición. AlfaOmega – Ra-Ma. Sznajdleder P. A. (2010). Java a fondo. Estudio del lenguaje y desarrollo de aplicaciones. Alfaomega.



Unidad de Aprendizaje	Contenido Temático	Referencias
	4.2. Interfaces Gráficas de Usuario 4.2.1 Contenedores 4.2.2 Componentes 4.2.3 Eventos	Wu Thomas. (2008). Programación en JAVA. Introducción a la programación orientada a objetos. McGrawHill
	4.3 Flujos de Entrada/Salida 4.3.1 Texto 4.3.2 Binario 4.3.3 Serializable	Hubbard J. R. (2012). Programming with Java. Schaum's OutLines. Mc Graw Hill.

Unidad de Aprendizaje	Contenido Temático	Referencias
5. Tópicos Avanzados de la Programación	5.1. Recursividad 5.1.1. tipos de recursividad 5.1.2. Ventajas y desventajas	Joyanes Aguilar Luis, Z. M. (2011). programación en JAVA 6 Algoritmos, programación orientada a objetos e interfaz gráfica de usuario. México: Mc Graw Hill. Wu Thomas. 2008. Programación en JAVA. Introducción a la programación orientada a objetos. McGrawHill

8. ESTRATEGIAS, TÉCNICAS Y RECURSOS DIDÁCTICOS

Estrategias y técnicas didácticas	Recursos didácticos
<p>Estrategias de aprendizaje:</p> <ul style="list-style-type: none"> • Lectura y comprensión, • Reflexión, • Comparación. <p>Estrategias de enseñanza:</p> <ul style="list-style-type: none"> • ABP, • Aprendizaje activo, • Aprendizaje cooperativo, • Aprendizaje colaborativo, <p>Ambientes de aprendizaje:</p> <ul style="list-style-type: none"> • Aula, • Laboratorio, • Simuladores. <p>Actividades y experiencias de aprendizaje:</p> <ul style="list-style-type: none"> • Asistencia a congresos. <p>Técnicas</p> <ul style="list-style-type: none"> • Grupales, • De debate, 	<p>Materiales:</p> <ul style="list-style-type: none"> • Proyectors, • TICs, • Plumón y pizarrón, • Ejercicios • Libros, fotocopias • Artículos científicos • Antologías • Materiales audiovisuales • Programas informáticos (CD u on-line) educativos. • Aplicaciones Multimedia • Páginas Web • Webquest • Correo electrónico • Chats • Foros • Links





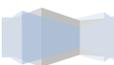
Estrategias y técnicas didácticas	Recursos didácticos
<ul style="list-style-type: none"> • Del diálogo, • De problemas, • De estudio de casos, • Cuadros sinópticos, • Mapas conceptuales, • Para el análisis, • Comparación, • Síntesis, • Mapas mentales, • Lluvia de ideas, • Analogías, • Portafolio, • Exposición. 	

9. EJES TRANSVERSALES

Eje (s) transversales	Contribución con la asignatura
Formación Humana y Social	Tomar consciencia de que los conocimientos y habilidades adquiridos pueden ser usados para modelar problemas de la vida real.
Desarrollo de Habilidades en el uso de las Tecnologías de la Información y la Comunicación	Interactuar con diferentes plataformas y herramientas que permitan integrar componentes en la solución de problemas.
Desarrollo de Habilidades del Pensamiento Complejo	Modelado de problemas de diversas disciplinas de la ciencia a través de componentes y sus relaciones.
Lengua Extranjera	Lectura de artículos especializados en el área de la programación orientada a objetos.
Innovación y Talento Universitario	Aplicación de los conceptos aprendidos en la solución de problemas de la vida real.
Educación para la Investigación	Capacidad de migrar el modelado de la solución de un problema a diversos lenguajes de programación orientados a objetos.

10. CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
• Tareas	10 %
• Prácticas de laboratorio	15 %
• Proyecto final	20 %
• Pruebas objetivas	30 %
• Participación en clase	10 %
• Asistencia	5 %
• Entregas puntuales	5 %
• Presentación de trabajos	5 %
Total	100%





11. REQUISITOS DE ACREDITACIÓN

Estar inscrito como alumno en la Unidad Académica en la BUAP
Asistir como mínimo al 80% de las sesiones para tener derecho a exentar por evaluación continua y/o presentar el examen final en ordinario o extraordinario
Asistir como mínimo al 70% de las sesiones para tener derecho al examen extraordinario
Cumplir con las actividades académicas y cargas de estudio asignadas que señale el PE

Notas:

- La entrega del programa de asignatura con sus respectivas actas de aprobación, deberá realizarse en formato electrónico, vía oficio emitido por la Dirección o Secretaría Académica a la Dirección General de Educación Superior.
- La planeación didáctica deberá ser entregada a la coordinación de la licenciatura en los tiempos y formas acordados por la Unidad Académica.

