

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA

COORDINACIÓN DE FORMACIÓN BÁSICA
COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA
PROGRAMA DE UNIDADES DE APRENDIZAJE POR COMPETENCIAS

I. DATOS DE IDENTIFICACIÓN

1. Unidad Académica: **Facultad de Ciencias**

2. Programa (s) de estudio: (Técnico, Licenciatura) **Licenciatura en Ciencias Computacionales** 3. Vigencia del plan: **2008-1**

4. Nombre de la Asignatura: **Programación Orientada a Objetos** 5. Clave: 9824

6. HC: **2** HL:**2** HT:**2** HE:**0** CR: **8**

7. Ciclo Escolar: **2009-1** 8. Etapa de formación a la que pertenece: **Disciplinaria**

9. Carácter de la Asignatura: **Obligatoria** Optativa

10. Requisitos para cursar la asignatura: **Introducción a la programación**

Formuló: **MC David Abdel Mejía Medina**

VoBo. **Marcelo Rodriguez Meraz**
Cargo: **SUBDIRECTOR**

II. PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE

La orientación a objetos ofrece mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software: la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas.

El propósito general del curso es diseñar aplicaciones que transporten objetos reales a abstracciones de código, relacionando estos objetos por medio de asociaciones y/o generalizaciones, entre otras relaciones, e implementándolos a través del uso de algún lenguaje orientado a objetos.

III. COMPETENCIA (S) DE LA UNIDAD DE APRENDIZAJE

Analizar y solucionar un problema mediante el desarrollo de aplicaciones orientadas a objetos en colaboración con un equipo de trabajo que valore un conjunto de soluciones propuestas en forma objetiva y honesta.

IV. EVIDENCIA (S) DE DESEMPEÑO

- Exposición de las soluciones desarrolladas para problemas reales e hipotéticos ante una audiencia y considerar los comentarios realizados a dicha solución.
- Documentar de manera formal las soluciones desarrolladas.
- Instrumentación de las soluciones desarrolladas en un lenguaje de programación orientado a objetos

Formatted: Portuguese (Brazil)

Formatted: Portuguese (Brazil)

Formatted: Font: 11 pt, Portuguese (Brazil)

V. DESARROLLO POR UNIDADES

Nombre de la unidad:

I. Introducción.

Competencia:

Comparar la programación orientada a objetos dentro del ámbito de las técnicas de programación en base a las aplicaciones típicas en las que es utilizada esta técnica.

Contenido temático

1. Evolución de la programación [1 horas].
2. Tecnología [1 horas].
3. Técnicas de programación [4 horas].
 - a. Estructurada.
 - b. Procedimental.
 - c. Orientada a Objetos.

Duración (6 horas)

Nombre de la unidad: II. Conceptos Orientados a Objetos	Competencia: Comprender la importancia de la técnica de Orientación a Objetos definiendo sus principales características que le permitan al estudiante la selección de este paradigma de programación.
Contenido temático 1. Objeto [6]. 2. Clase [3]. 3. Mensajes [2] 4. Ventajas de Orientación a Objetos [1].	Duración (12 horas)

Nombre de la unidad: III. Desarrollo Orientado a Objetos	Competencia: Plantear soluciones a problemas hipotéticos utilizando los conceptos del desarrollo orientado a objetos.
Contenido temático 1. Abstracción [2 horas]. 2. Encapsulamiento [2 horas]. 3. Polimorfismo [2 horas]. 4. Herencia [4 horas]. 5. Híbridos [4 horas]. 6. Relaciones [4 horas].	Duración (18 horas)

Nombre de la unidad: IV. Programación Orientada a Objetos.	Competencia: Utilizar los conceptos de Orientación a objetos en un lenguaje de programación que soporte este paradigma que le permitan obtener la experiencia en el desarrollo de aplicaciones orientadas a objetos.
Contenido temático 1. Creación de clases [2 horas]. 2. Encapsulamiento [2 horas]. 3. Polimorfismo [4 horas]. 4. Herencia Simple [4 horas]. 5. Herencia Múltiple [4 horas]. 6. Paquetes [4 horas]. 7. Reutilización de código [4 horas].	Duración (28 horas)

Nombre de la unidad: V. Aplicaciones.	Competencia: Aplicar los conceptos avanzados de la programación orientada a objetos en la solución de problemas de la vida real a través de la implementación de un sistema de cómputo.
Contenido temático 1. Interfaces de usuario [8 horas]. 2. Colecciones de objetos [4 horas]. 3. Flujos [4 horas]. 4. Excepciones [2 horas]. 5. Hilos [6 horas]. 6. Trabajo en red [8 horas]	Duración (32 horas)

VI. ESTRUCTURA DE LAS PRÁCTICAS

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
1.	El alumno conocerá la posición de la programación orientada a objetos dentro del ámbito de las técnicas de programación en base a las aplicaciones típicas en las que es utilizada esta técnica	Los alumnos, organizados en pequeños grupos, realizarán una investigación sobre las técnicas de programación (incluyendo un ejemplo, los lenguajes, y las ventajas y desventajas). En forma individual se realizará un reporte sobre la situación de la programación orientada a objetos en la actualidad.	Libros Internet	1 semana para investigación 10 minutos para la presentación 1 día para el reporte
2.	El alumno identificará las principales características de la técnica de Orientación a Objetos evaluando las causas por las que se vuelve importante el paradigma de la programación orientada a objetos.	El alumno realizará ejercicios sobre la definición de conceptos y su relación con la vida real. Destacará las diferencias en utilizar estos conceptos en la vida diaria y desde el punto de vista de computación.	Libros Internet Papel	Ejercicios diarios de 15 minutos.
3.	El alumno utilizará los conceptos del desarrollo orientado a objetos para plantear una solución de problemas hipotéticos.	El alumno desarrollará una solución a un problema hipotético planteado por el profesor. En este se deberán destacar los componentes necesarios para la solución así como un modelo para visualizar dicha solución. Estos modelos deberán ser expuestas al grupo.	Papel Diapositivas	1 semana con revisiones por clase. 10 minutos de exposición.
4.	El alumno dominará una técnica para implementar los conceptos de Orientación a objetos en un lenguaje de programación	El alumno desarrollará un sistema de cómputo en el cual se visualice el modelo desarrollado para establecer la solución de un problema.	Internet Computadora Lenguaje de programación	2 semanas para el desarrollo con revisiones por clase. 15 minutos para exposición.
5.	El alumno identificará problemas reales y desarrollará una solución del mismo desde la perspectiva de orientación a objetos.	El alumno planteará un problema real observado y desarrollará una solución. Dicha solución será desarrollada en papel e implementada en algún lenguaje de programación.	Papel Computadora Lenguaje de programación	2 semanas para el planteamiento. 1 semana para el desarrollo.

VI. METODOLOGIA DE TRABAJO

- Las clases en el aula se impartirán en el horario establecido por el centro, en ellas el profesor expondrá los contenidos del temario de la asignatura. En las primeras lecciones del curso se concentrará un máximo número de conceptos teóricos. La distribución de los temas determina una metodología de enseñanza cíclica, en la que los mismos conceptos se abordan de forma recurrente a lo largo del curso.
- Se utilizarán tres tipos de clases: las de teoría, las de problemas y las de laboratorio o prácticas. Las clases de teoría son aquellas en las que el profesor expone los principales conceptos teóricos. Las clases de problemas se centran en la resolución de ejercicios y ejemplos. Las clases de prácticas se realizarán en el laboratorio, en estas clases cada alumno tendrá que implementar en un lenguaje de programación orientado a objetos, los ejercicios propuestos.
- Los alumnos, al final del semestre, realizan trabajos sobre problemas reales.
- Normalmente los trabajos se desarrollan en equipos solicitando reportes de los resultados obtenidos y las observaciones realizadas.

VII. CRITERIOS DE EVALUACION

- Exámenes por unidad. Después de cada unidad el maestro realizará un examen. Después del segundo examen se considera que el estudiante no ha olvidado sus conocimientos y probablemente se utilice información vistas en exámenes pasados.
- Tareas por semana. Estas tareas tienen un nivel medio de complejidad, por tal razón merecen un porcentaje dentro de la evaluación.
- Ejercicios en clase. Tareas diarias.
- Proyecto final. Es un sistema computacional que utilice los conceptos vistos en clase en la resolución de un problema de la vida real.

VIII. BIBLIOGRAFIA

Básica

- Booch, Grady, *Análisis y Diseño Orientado a Objetos: con aplicaciones*, 2da Edición, Addison-Wesley, 1996, ISBN: 9684443528
- Hunt, John, *Java and Object Orientation: an introduction*, 2da Edición, Springer, 2002, ISBN: 1852335696
- Wu, C. Thomas, *An introduction to object-oriented programming with Java*, McGraw-Hill, 2004, ISBN: 0073040959
- Morelli, Ralph, *Java, Java, Java!: object-oriented problem solving*, Prentice Hall, 2003, ISBN: 0130333700
- Kak, Avinash C., *Programming with objects: a comparative presentation of object-oriented programming with C++ and Java*, John Wiley, 2003, ISBN: 0471268526
- Schach, Stephen R., *An introduction to object-oriented systems análisis and design with UML and the unified process*, McGraw-Hill, 2004, ISBN: 0072826460
- Booch, Grady, James Rumbaugh, Ivar Jacobson, *El lenguaje Unificado de Modelado*, Addison-Wesley, 1999, ISBN: 84-7829-028-1
- Budd, T. *Introducción a la programación orientada a objetos*. Addison-Wesley. 1989.
- Savitch. *Problem solving with C++*. Addison Wesley. 1998
- Holzner, S. *La biblia de Java 2*. Anaya Multimedia. 2000

Complementaria

- Khoshafian, S.; Abnous, R. “*Object-orientation, Concepts, Languages, Databases, User Interfaces*”. John Wiley & Sons, Inc., 1990.
- Winblad, A. L.; Edwards, S. D.; King, D. R. “*Object-Oriented Software*”. Addison-Wesley Publishing Company. 1990.

Formatted: French (France)

Formatted: Font: Not Italic, French (France)