

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**COORDINACIÓN DE FORMACIÓN BÁSICA**  
**COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN**  
**UNIVERSITARIA**  
**DEPARTAMENTO DE ACTUALIZACIÓN CURRICULAR Y FORMACIÓN**  
**DOCENTE**

**DESCRIPCIÓN GENÉRICA DE UNIDADES DE APRENDIZAJE**

Descripción Genérica

Nombre: **Programación Orientada a Objetos**  
 Etapa: **Disciplinaria**  
 Área de conocimiento: **Programación e Ingeniería de Software**

**Competencia:**

Desarrollar y analizar soluciones a problemas reales o hipotéticos en colaboración con un equipo de trabajo, valorando las soluciones propuestas en forma objetiva y honesta para resolverlos mediante la programación orientada a objetos.

**Evidencia de desempeño:**

- Prácticas de laboratorio y tareas extraclase
  - Ejercicios que involucren la solución de problemas reales e hipotéticos siguiendo el enfoque de la orientación a objetos.
  - Exposición de las soluciones desarrolladas ante una audiencia y considerar los comentarios realizados a dicha solución.
  - Documentación formal de las soluciones desarrolladas.
  - Instrumentación de las soluciones desarrolladas en un lenguaje de programación.
- Exámenes teóricos y prácticos
- Proyecto final

Distribución	HC	HL	HT	HPC	HCL	HE	CR	Requisito
	2	2	2				8	<b>Recomendadas: Diseño de Algoritmos e Introducción a la Programación</b>

## Contenidos Temáticos

1. Introducción
  - 1.1. Evolución de la programación
  - 1.2. Paradigmas de programación
  - 1.3. Tipos de lenguajes de programación
  
2. Conceptos de la Orientación a Objetos
  - 2.1. Objetos
  - 2.2. Clases e Instancias de clases
  - 2.3. Mensajes
  - 2.4. Ventajas de la Orientación a Objetos
  
3. Desarrollo Orientado a Objetos
  - 3.1. Abstracción
  - 3.2. Encapsulamiento
  - 3.3. Jerarquías de clases y objetos
    - 3.3.1. Jerarquía Padre/Hijo, la herencia
    - 3.3.2. Jerarquía Todo/Parte, la agregación
  - 3.4. Relaciones entre clases y objetos
    - 3.4.1. Dependencia (relaciones de uso)
    - 3.4.2. Generalización (relaciones de herencia)
    - 3.4.3. Asociación
    - 3.4.4. Agregación simple
    - 3.4.5. Composición
  - 3.5. Visibilidad y alcance de atributos y operaciones
  - 3.6. Polimorfismo
  
4. Programación Orientada a Objetos
  - 4.1. Creación de clases y objetos
  - 4.2. Encapsulamiento
  - 4.3. Polimorfismo
  - 4.4. Herencia Simple
  - 4.5. Herencia Múltiple
  - 4.6. Paquetes
  - 4.7. Reutilización de Código
  
5. Aplicaciones
  - 5.1. Interfases de usuario
  - 5.2. Colecciones de objetos
  - 5.3. Flujos
  - 5.4. Excepciones
  - 5.5. Hilos
  - 5.6. Trabajo en red

### **Referencias bibliográficas actualizadas**

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