



UNIVERSIDAD MICHOACANA DE
SAN NICOLÁS DE HIDALGO
FACULTAD DE INGENIERÍA ELÉCTRICA



Nombre de la materia:	Programación de Computadoras II
Clave:	CI0002-T
No. de horas/semana:	4
Total de horas:	64
No. de créditos:	8
Prerrequisitos:	Programación de Computadoras (CI0000-T)

Objetivo general: Que el alumno mejore su capacidad de desarrollo de software así como su productividad al producir código fácil de mantener y altamente reutilizable, aprovechando las ventajas del encapsulamiento de datos, la herencia, la abstracción de datos y demás ventajas que ofrece la programación orientada a objetos. Al finalizar, el alumno tendrá los conocimientos necesarios para aprobar el examen SCJP.

Programa sintético

1. Introducción	4 hrs.
2. Datos y Expresiones	4 hrs.
3. Uso de Clases y Objetos	6 hrs.
4. Control de Flujo	4 hrs.
5. Examen	2 hrs.
6. Modelado de Objetos	8 hrs.
7. Diseño de Clases en Java	14 hrs.
8. Examen	2 hrs.
9. Uso de Collections y Generics	4 hrs.
10. Concurrencia	8 hrs.
11. Java I/O	6 hrs.
12. Examen	2 hrs.
Total: 64 hrs.	

Programa desarrollado

1. Introducción	4 hrs.
1.1 Propiedades deseables: Reusabilidad y extensibilidad, entre otras.	
1.2 Programación orientada a objetos.	
1.3 Ingeniería de Software. Ventajas de la Programación Orientada a Objetos.	
1.4 El lenguaje de programación Java	
1.5 Ciclo de desarrollo de programas en Java	



- 2. Datos y Expresiones 4 hrs.
 - 2.1 Tipos de datos en java (primitivos, Clases, Interfaces, Enumeraciones)
 - 2.2 Variables y asignación
 - 2.3 Cadenas de caracteres
 - 2.4 Expresiones
 - 2.5 Conversión de datos
 - 2.6 Clases wrapper
- 3. Uso de Clases y Objetos 6 hrs.
 - 3.1 La API de Java
 - 3.2 Creación de objetos
 - 3.3 La clase String
 - 3.4 Paquetes
 - 3.5 La clase Random
 - 3.6 La clase Math
 - 3.7 Formateo de Salida
- 4. Control de Flujo 4 hrs.
 - 4.1 Sentencia if
 - 4.2 Comparando datos
 - 4.3 Sentencia switch
 - 4.4 Sentencia while
 - 4.5 Sentencias do y for
 - 4.6 Iteradores
 - 4.7 Recursividad
 - 4.8 Assertions
 - 4.9 Introducción a Excepciones
- 5. Examen 2 hrs.
- 6. Modelado de Objetos 8 hrs.
 - 6.1 Modelos de especificación del Dominio: La técnica CRC.
 - 6.2 Diagrama de Clases (modelo conceptual)
 - 6.3 Ejemplos prácticos para descubrir Clases y Métodos utilizando la Técnica CRC.
 - 6.4 Utilizando la Técnica CRC, encontrar responsabilidades para las Clases y
 - 6.5 Introducción a especificación de modelos con el Lenguaje de Modelado Unificado (UML). Tipos de Diagramas UML Diagramas de clases. Elementos del Diagrama de Clases. Relaciones entre clases: Asociación, Composición, Agregación, Dependencia, Herencia, Ejemplos de relaciones entre clases



7. Diseño de Clases en Java 14 hrs.
- 7.1 Repaso de Clases y Objetos
 - 7.2 Anatomía de una Clase (java y UML)
 - 7.3 Encapsulación
 - 7.4 Anatomía de un método
 - 7.5 Revisión de constructores
 - 7.6 Miembros estáticos
 - 7.7 Relaciones entre clases
 - 7.8 Interfaces
 - 7.9 Herencia
 - 7.10 Subclases
 - 7.11 Sobreescritura de métodos
 - 7.12 Visibilidad
 - 7.13 Polimorfismo
 - 7.14 Polimorfismo usando herencia
 - 7.15 Polimorfismo usando Interfaces
 - 7.16 Revisión de Excepciones
8. Examen 2 hrs.
9. Uso de Collections y Generics 4 hrs.
- 9.1 Uso de == y del método equal, y sobreescritura del método hashCode
 - 9.2 Clases de la API Collections (Set, List, Map, etc)
 - 9.3 Generics
 - 9.4 Uso de java.util.Comparator and java.lang.Comparable así como el paquete java.util para escribir código que ordene y realice búsquedas.
10. Concurrencia 8 hrs.
- 10.1 Creación de hilos a partir de java.lang.Thread y java.lang Runnable.
 - 10.2 Estados de un hilo
 - 10.3 Solución de problemas de acceso concurrente
 - 10.4 Uso de wait, notify, or notifyAll.
11. Java I/O 6 hrs.
- 11.1 Uso de BufferedReader, BufferedWriter, File, FileReader, FileWriter and PrintWriter.
 - 11.2 Serialización de objetos usando las siguientes clases: DataOutputStream, FileInputStream, FileOutputStream, ObjectInputStream, ObjectOutputStream and Serializable.
 - 11.3 Uso de la clase java.util.Locale
 - 11.4 Uso de expresiones regulares



12. Examen 2 hrs.

Bibliografía básica:

- Paul Deitel and Hayvey M. Deitel. "Java How to Program". Ninth edition. Prentice Hall. 2011.
- Katherine Sierra, Bert Bates. "SCJP Sun Certified Programmer for Java 6 Exam 310-065", McGraw-Hill Osborne Media; 1a edición (Junio 24, 2008).
- Danny Poo & Derek Kiong. Object Oriented Programming and Java. Springer 2008.
- C. Thomas Wu. Introducción a la programación Orientada a Objetos con Java. Mc Graw Hill. 2001.
- Lewis & Loftus. "Java Software Solutions: Foundations of Program Design", Addison-Wesley, 6a Edición.

Metodologías de enseñanza-aprendizaje:

- Revisión de conceptos, análisis y solución de problemas en clase (X)
- Lectura de material fuera de clase (X)
- Ejercicios fuera de clase (tareas) (X)
- Elaboración de reportes técnicos o proyectos (X)

Metodologías de evaluación:

- Asistencia (X)
- Tareas (X)
- Exámenes de academia o departamentales (X)

Revisores:

Programa modificado por Dr. Félix Calderón Solorio. Mayo de 2017.