



UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE CIENCIAS
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACION

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	INTRODUCCION A LA PROGRAMACION ORIENTADA A OBJETOS
CÓDIGO	:	CC201
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC102 – INTRODUCCION A LA PROGRAMACIÓN
CONDICIÓN	:	OBLIGATORIO
HORAS POR SEMANA	:	06 (TEORÍA: 03, PRÁCTICA: 03)
SISTEMA DE EVALUACIÓN	:	G

OBJETIVO

Introducir al alumno a los fundamentos del paradigma de orientación a objetos, permitiendo o asimilar los conceptos necesarios para desarrollar un sistema de información.
Describir el modelo OO
Experimentar como las técnicas OO favorecen el desarrollo de software de calidad, analizando sobre todo cómo facilitan la reutilización y extensibilidad.
Contrastar como diferentes lenguajes de programación OO (Java, C++) reflejan los conceptos del paradigma.
Enseñar un lenguaje OO (en este caso, principalmente C++ y Java)
Enseñar técnicas de programación OO
Presentar ejemplos de aplicación.

PROGRAMA ANALÍTICO

1. Introducción a la programación orientada a objetos

I: Historia de C++. II: Historia de Java. III: Internet y la Web. IV: Entorno de Java.

2. Aritmética y operadores

I: Aritmética en Java. II: Operadores de igualdad y relacionales. III: I/O básico.

3. Applets

I: SDK de java. II: Dibujando cadenas y líneas.

4. Instrucciones de control

I: Algoritmos. II: Pseudo código. III: Estructuras de control (if, if else, while). IV: Operadores de asignación compuestos. V: Operadores de incremento y decremento. VI: Tipos primitivos. VII: Instrucciones de repetición (for, do while, switch, break, continue). VIII: Operadores lógicos.

5. Métodos

I: Introducción. II: Métodos de la clase Math. III: Declarando métodos y el paso de argumentos. IV: Algunos métodos de la API de java. V: Recursividad vs. iteración.

6. Arreglos (arrays)

I: Creación y declaración de arreglos. II: Referencia y parámetros de referencia. III: Pasando arreglos a un método. IV: Ordenamiento, búsqueda. V: Arreglos multidimensionales.

7. Creando mis propias clases y objetos

I: Implementando tipos abstractos en una clase. II: Alcance de una clase. III: Constructores, constructores sobrecargados (overloaded). IV: Set y get. V: Recolección de basura. VI: Static, final. VII: Paquetes. VIII: Reutilización de código, abstracción de datos, encapsulamiento.

8. Herencia

I: Superclases y subclases. II: Miembros protected. III: Constructores y finalizadores de las subclases.

9. Polimorfismo

I: Relación entre objetos en una jerarquía de herencia. II: Clases y métodos abstractos. III: Constructores y finalizadores de las subclases. IV: Final. V: Clases anidadas.

10. Cadenas (strings) y caracteres

I: Definiciones. II: Clase String. III: Clase StringBuffer. IV: Clase Character. V: Clase StringTokenizer. VI: Clases Pattern y Matcher.

11. Creando una Interfaz gráfica de usuario (GUI)

I: Swing. II: JLabel, campos de texto. III: Eventos y manejo de eventos. IV: Botones, check boxes, radio buttons, combo boxes, listas.

12. Manejo de excepciones

I: Generalidades. II: Ejemplo: división entre cero. III: Jerarquía de excepciones. IV: Finally. V: Limpieza de la pila. VI: printStackSize, getStackSize, getMessage. VII: Declarando nuevos tipos de excepciones.

13. Archivos y flujos

I: Introducción. II: Clase File. III: Archivos de acceso secuencial. IV: Archivos de acceso aleatorio.

14. Estructuras dinámicas de datos

I: Introducción. II: Asignación dinámica de memoria. III: Listas enlazadas. IV: Pilas. V: Colas. VI: Árboles.

BIBLIOGRAFÍA

1. Deitel, H. y Deitel, P., “Como programar en Java”, 5ta. Edición, Pearson Educación, 2004
2. Lafore R., “Object-oriented programming in C++”, 4ta. Edición, 2002, SAMS
3. Poo, K., Kiong, P., Ashok, S., “Object oriented programming and Java”, 2da. Edición, 2008, Springer
4. Preiss, B., “Data Structures and Algorithms with Object-Oriented Design Patterns in Java”, 1ra. Ed., 2001, <http://www.brpreiss.com/books/opus5/html/book.html>Thompson, S. The Craft of Functional Programming, 2E. Addison Wesley, 1999.