SÍLABO

Programación Orientada a Objetos

I. INFORMACIÓN GENERAL

CÓDIGO	:	CC211
CICLO	:	3
CREDITOS	:	4
HORAS POR SEMANA	:	6 (Teoría: 3, Práctica: 3)
PREREQUISITO	:	Fundamentos de Programación
CONDICIÓN	:	Obligatorio
PROFESOR	:	Juan Espejo
E-MAIL	:	jespejod@uni.edu.pe

II. SUMILLA El curso introduce al estudiante los fundamentos del paradigma de orientación a objetos (OO), permitiendo que asimile los conceptos necesarios para desarrollar un sistema de información. Se describe el modelo OO para que el estudiante experimente cómo las técnicas OO favorecen el desarrollo de software de calidad, analizando sobre todo cómo facilitan la reutilización y extensibilidad. Finalmente, los estudiantes programan bajo el paradigma OO en un lenguaje en particular (en este caso, Java).

III. COMPETENCIAS Al finalizar la asignatura, el estudiante:

1) Complementa la programción entructurada en la programación OO.
2) Emplea pensamiento abstracto para el diseño de interfaces.
3) Utiliza constructores en la implementación de todas sus clases.
4) Utiliza la sobrecarga de métodos: en particular constructores múltiples.
5) Domina y aplica las técnicas fundamentales de la programación OO como: encapsulamiento y ocultamiento de datos, herencia, polimorfismo y composición.
6) Modela sistemas del mundo real a través de diseños OO.
7) Maneja diagramas de UML para el diseño de sus clases
8) Maneja errores posibles que pueden surgir durante la ejecución del programa.
9) Emplea la reutilización de código para la implementación de nuevos diseños.

IV. UNIDADES DE APRENDIZAJE

1. **INTRODUCCIÓN A LOS CONCEPTOS ORIENTADO A OBJETOS / 3 HORAS**
   Los conceptos fundamentales / Objetos y sistemas heredados / Programación por procedimientos versus programación orientada a objetos / ¿Qué es exactamente un objeto? / ¿Qué es exactamente una clase? / Encapsulamiento y ocultación de datos / Herencia / Polimorfismo / Composición.[5]

2. **CÓMO PENSAR EN TÉRMINOS DE OBJETOS / 3 HORAS**
   Conociendo la diferencia entre la interfaz y la implementación / Usando pensamiento abstracto al diseñar interfaces / Proporcionando la mínima interfaz de usuario posible.[5]

3. **COMENZANDO A PROGRAMAR EN JAVA / 3 HORAS**
   Ingresando y mostrando información / Operadores fundamentales / Variables de instancia, métodos *set* y métodos *get* / Tipos primitivos versus tipos de referencia / Inicializando objetos con constructores / usando cuadros de diálogos.[2]

4. **FUNDAMENTOS DE LA PROGRAMACIÓN EN JAVA / 6 HORAS**
   Estructuras de control selectivas, estructuras de control repetitivas y anidamiento / Métodos `static`, atributos `static` y clase `Math` / Declarando métodos con múltiples parámetros / Conversión de tipos / Paquetes de la API Java / Declarando y creando arreglos / Pasando arreglos a métodos / Paso por valor versus paso por referencia / Arrreglos multidimensionales / Clase `Arrays` / Fundamentos de caracteres y cadenas / Clase `String` / Clase `StringBuilder` / Archivos y transmisiones / Usando clases NIO e interfaces para obtener archivos e información de directorio / Acceso secuencial de archivos de texto / Objeto serialización / Abriendo archivos con `JFileChooser`.[2][4]

5. **CONCEPTOS AVANZADOS DE ORIENTACIÓN A OBJETOS / 3 HORAS**
   Constructores / Manejo de Errores / La importancia del alcance / Sobrecarga del operador / Herencia múltiple / Operaciones de objetos.[5]

6. **LA ANATOMÍA DE UNA CLASE / 3 HORAS**
   El nombre de una clase / Comentarios / Atributos / Constructores / Accesorios / Métodos de interfaz pública / Métodos de implementación privada.[5]

7. **DIRECTRICES DE DISEÑO DE UNA CLASE / 3 HORAS**
   Modelando sistemas del mundo real / Identificando las interfaces públicas / Diseñando constructores robustos / Diseñando manejo de errores en una clase / Diseñando con reutilización en mente / Diseñando con extensibilidad en mente / Diseñando con capacidad de mantenimiento en mente / Usando persistencia de objetos.[5]

8. **DISEÑANDO CON OBJETOS / 3 HORAS**
   Directrices de diseño: realizando el análisis apropiado, reuniendo los requeri-

mientos, desarrollando un prototipo de una interfaz de usuario, identificando las clases / Empaquetador de objetos: código estructurado, código estructuras empaquetado y código no portable estructurado.[5]

9. **DOMINANDO LA HERENCIA Y LA COMPOSICIÓN / 3 HORAS**
   Reutilizando objetos / Herencia / Composición / El porqué encapsulamiento es funsamental para orientación a objetos: cómo la herencia debilita el encapsulamiento, responsabilidad del objeto, clases abstractas, métodos virtuales y protocolos.[5]

10. **CLASES Y OBJETOS EN JAVA / 6 HORAS**
    Controlando el acceso a miembros / Refirindose a los miembros del objeto con la referencia `this` / Constructores por defecto y sin argumento / Composición / Tipos `enum` / Recolector de basura / Miembros de clase `static`.[1]

11. **HERENCIA EN JAVA / 6 HORAS**
    Superclases y subclases / Miembros `protected` / Relación entre superclases y subclases / Constructores en subclases.[2][1]

12. **POLIMORFISMO E INTERFACES EN JAVA / 6 HORAS**
    Ejemplos de polimorfismo / Demostrando comportamiento polimórfico / Clases y métodos abstractos / Caso de estudio: sistema de nómica usando polimorfismo / Asignaciones permitidas entre variables de superclase y subclase / Creando y usando interfaces / Mejoras de la interfaz de Java SE 8.[2][1]

13. **MANEJO DE EXCEPCIONES EN JAVA / 3 HORAS**
    Ejemplos / Cuando usar el manejo de excepciones / Jerarquía de excepciones de Java / Bloque `finally` / Excepciones encadenadas / Declarando nuevos tipos de excepciones / Precondiciones y postcondiciones / Aserciones.[2]

14. **MARCOS Y REUTILIZACIÓN / 3 HORAS**
    Código: ¿Reutilizar o no reutilizar? / ¿Qué es un marco? / ¿Qué es un contrato? / Un ejemplo de negocio electrónico.[2][1]

V. CALENDARIO ACADÉMICO

Semana	Fecha	Actividad
1	18/03/19 – 23/03/19	Prueba de entrada / Desarrollo de la prueba de entrada / Unidad 1 / Lectura: capítulos 1-2 de [5], apéndice 2 de [4] y capítulos 2-5 de [2]
2	25/03/19 – 30/03/19	Unidad 2 / Unidad 3 / Lectura: capítulo 6-7 de [2], capítulo 1 de [3] y capítulo 3 de [5]
3	01/04/19 – 06/04/19	Unidad 4 / Lectura: capítulos 14-15 de [2], capítulo 1 de [1] y capítulo 1 de [4]
4	08/04/19 – 13/04/19	**Práctica Calificada 1** / Preguntas sobre la PC1 / Unidad 5 / Lectura: capítulos 4-5 de [5]
5	15/04/19 – 20/04/19	Unidad 6 / Unidad 7 / Lectura: capítulos 6-7 de [5]
6	22/04/19 – 27/04/19	Unidad 8 / Unidad 9 / Lectura: capítulo 8 de [2]
7	29/04/19 – 04/05/19	**Práctica Calificada 2** / Preguntas sobre la PC2 / Unidad 10 (1/2) / Lectura: capítulo 2 de [1] y capítulo 2 de [4]
8	06/05/19 – 11/05/19	**Examen Parcial**
9	13/05/19 – 18/05/19	Unidad 10 (2/2) / Laboratorio 1 / Laboratorio 1 / Lectura: capítulo 9 de [2] y capítulo 3 de [1]
10	20/05/19 – 25/05/19	Expociencia
11	27/05/19 – 01/06/19	Unidad 11 / Lectura: capítulo 2 de [3] y capítulo 2 de [4]
12	03/06/19 – 08/06/19	Laboratorio 2 / **Práctica Calificada 3** / Preguntas sobre la PC3 / Lectura: capítulo 10 de [2]
13	10/06/19 – 15/06/19	Unidad 12 / Lectura: capítulo 3 de [4]
14	17/06/19 – 22/06/19	Laboratorio 3 / **Práctica Calificada 4** / Preguntas sobre la PC4 / Lectura: capítulo 11 de [2]
15	24/06/19 – 29/06/19	Unidad 13 / Unidad 14 / Lectura del capítulo 8 de [5]
16	01/07/19 – 06/07/19	**Examen Final**
17	08/07/19 – 12/07/19	Firma de acta de compendio de prácticas
18	15/07/19 – 20/07/19	**Examen Sustitutorio**

VI. METODOLOGÍA

Se desarrollan sesiones de teoría, exposiciones y laboratorio de cómputo. En las sesiones de teoría, el docente presenta los conceptos, la lógica del pensamiento OO y aplicaciones. En las sesiones de exposiciones, los estudiantes presentan sus desarrollos –en el lenguaje de programación Java– de las las diversas aplicaciones presentadas en teoría, se analiza cada presentación de manera colaborativa y al final el docente da un 'feedback' sobre la exposición. En las sesiones de laboratorio se usa el IDE NetBeans 8.2 para resolver problemas y analizar su solución. Durante el curso el estudiante presenta y expone un proyecto integrador. En todas las sesiones se promueve la participación activa del estudiante.

VII. FÓRMULA DE EVALUACIÓN

Sistema de Evaluación G; es decir, el promedio final resulta de promediar las siguientes notas:

Pruebas de evaluación	Peso
Examen parcial	1
Examen final	1
Promedio de prácticas	1

Se toman seis prácticas, se elimina una.

Referencias

[1] Dathan, B., and Ramnath, S. *Object-Oriented Analysis, Design and Implementation: An Integrated Approach*, 2nd ed. Undergraduate Topics in Computer Science. Springer, 2015.

[2] Deitel, P., and Deitel, H. *Java How to Program: Early Objects*. Pearson Education, 2015.

[3] Freeman, E., Bates, B., Sierra, K., and Robson, E. *Head First Design Patterns: A Brain-Friendly Guide*. Head First. O'Reilly Media, 2004.

[4] McLaughlin, B., Pollice, G., and West, D. *Head First Object-Oriented Analysis and Design*. Head First. O'Reilly Media, 2006.

[5] Weisfeld, M. *The Object-Oriented Thought Process*, 4th ed. Developer's Library. Addison-Wesley Professional, 2013.