



## I. INFORMACIÓN GENERAL

<b>CURSO</b>	:	Programación Orientada A Objetos
<b>CÓDIGO</b>	:	IS04
<b>CICLO</b>	:	201000
<b>CUERPO ACADÉMICO</b>	:	<b>Contreras Chávez, Estanislao</b> <b>Montero Flores, Roberto Elías</b> <b>Murga Fernández, Raúl Jorge</b> <b>Subauste Oliden, Daniel Alejandro</b>
<b>CRÉDITOS</b>	:	5
<b>SEMANAS</b>	:	16
<b>HORAS</b>	:	3 H (Laboratorio) Semanal
<b>ÁREA O CARRERA</b>	:	Ingeniería de Sistemas - Epe

## II. MISIÓN Y VISIÓN DE LA UPC

Misión: Formar líderes íntegros e innovadores con visión global para que transformen el Perú.

Visión: Ser líder en la educación superior por su excelencia académica y su capacidad de innovación.

## III. INTRODUCCIÓN

Desde el nacimiento de los primeros lenguajes de Programación Orientados a Objetos (POO), la orientación a objetos (OO) se ha presentado como la solución a muchos de los problemas del desarrollo de software, consolidándose como el paradigma de desarrollo para sistemas de complejidad media o alta.

La POO es algo más que una colección de lenguajes de programación, tales como Java, C++, C#, VB.NET, etc. El desarrollo de programas OO implica la creación de modelos del mundo real y la construcción de programas o sistemas basados en esos modelos.

Tras la aceptación del paradigma OO como el más adecuado para producir software de calidad, los diversos métodos de desarrollo de software OO se unificaron y es así como nace el Lenguaje Unificado de Modelamiento (UML Unified Modeling Language). UML ha ejercido un gran impacto en la comunidad software, pues representa un paso más en la dirección de acercar el lenguaje de soluciones informáticas al lenguaje en que se plantean los problemas.

El curso presenta los conceptos fundamentales de POO y los alumnos los ponen en práctica usando el IDE Netbeans como herramienta de modelamiento UML así como de programación. Como lenguaje de programación se utiliza Java para la implementación de programas. El requisito básico para este curso es poseer conocimientos y experiencia en algún lenguaje de programación convencional, tales como el Pascal, C, Java, etc.

## IV. LOGRO (S) DEL CURSO

Al concluir el curso, el estudiante será capaz de construir aplicaciones bajo un enfoque de programación orientado a objetos cumpliendo con los principios y fundamentos de la programación orientada a objetos.

## V. UNIDADES DE APRENDIZAJE

### UNIDAD N°: 1 Conceptos fundamentales de POO.

#### LOGRO

Conoce y emplea los conceptos fundamentales de la POO en el análisis de problemas.

#### TEMARIO

- Programación orientada a objetos.
- Clases y objetos
- Atributos, estados, métodos y mensajes.
- Abstracción
- Encapsulamiento y Ocultamiento de la información.
- Principio Cliente-Servidor
- Jerarquía de Clases

#### HORA(S) / SEMANA(S)

Semana 1

### UNIDAD N°: 2 Relaciones entre clases.

#### LOGRO

- Diseña las clases que solucionan un problema y organiza dichas clases usando las relaciones de asociación, agregación, composición y generalización.
- Elabora diagramas de clases usando UML y Netbeans como herramienta de modelamiento.

#### TEMARIO

- UML
- Modelos y diagramas
- Relaciones entre clases: asociación, agregación, composición y generalización.
- Diagramas de clases
- Modelado en Netbeans

#### HORA(S) / SEMANA(S)

Semana 2 a 3

### UNIDAD N°: 3 Programación Orientada a Objetos con Java

#### LOGRO

El alumno construye aplicaciones en Java donde aplica los conceptos fundamentales de la POO y las construcciones básicas del lenguaje.

Desarrolla programas java con la calidad del código necesarias para aplicaciones orientadas a objetos y los construye bajo la importancia de un buen diseño.

#### TEMARIO

- Introducción a Java.
- Operadores y tipos de datos.
- Control del flujo de programa.
- Clases y objetos.

- Arreglos de objetos.

**HORA(S) / SEMANA(S)**

Semana 4 a 7

**UNIDAD N°: 4 HERENCIA****LOGRO**

Conoce y utiliza los conceptos de herencia y polimorfismo en el diseño de clases e implementación de programas.

**TEMARIO**

- Descubriendo el concepto herencia
- Superclases, subclasses.
- Generalización / Especialización de clases.
- Visibilidad de métodos y atributos en una jerarquía de clases.
- Tipos de herencia
- Herencias múltiples
- Herencia a través de interfaces
- Beneficios y costos de la herencia
- Reutilización de código en una jerarquía de clases.
- Implementación de herencia de clases en Java.

**HORA(S) / SEMANA(S)**

Semana 9 a 10

**UNIDAD N°: 5 POLIMORFISMO****LOGRO**

Conoce y utiliza los conceptos de herencia y polimorfismo en el diseño de clases e implementación de programas.

**TEMARIO**

- Clases abstractas y clases concretas
- Métodos abstractos
- Métodos redefinidos
- Polimorfismo
- Implementación de polimorfismo en una jerarquía de clases.

**HORA(S) / SEMANA(S)**

Semana 11 a 12

**UNIDAD N°: 6 INTERFACES Y MANEJO DE ERRORES****LOGRO**

Conoce y define interfaces como mecanismo de implementación de herencia múltiple, depura el código aprendido y lo perfecciona utilizando manejo de errores con excepciones.

**TEMARIO**

- Interfaces
- Uso de interfaces
- Manejo de errores

- Excepciones

**HORA(S) / SEMANA(S)**

Semana 13 a 14

**VI. METODOLOGÍA**

- Exposición de los conceptos teóricos con ayuda de transparencias y diapositivas.
- Prácticas dirigidas en salón orientado a la resolución de problemas.
- Prácticas calificadas en el laboratorio mediante trabajos individuales.
- Trabajo práctico final orientado a integrar el conocimiento y habilidades adquiridos

**VII. EVALUACIÓN**

**FÓRMULA**

20% (EA1) + 30% (EB1) + 15% (PC1) + 15% (PC2) + 20% (TF1)

TIPO DE NOTA	PESO %
EA - EVALUACIÓN PARCIAL	20
EB - EVALUACIÓN FINAL	30
PC - PRÁCTICAS PC	15
PC - PRÁCTICAS PC	15
TF - TRABAJO FINAL	20

**VIII. CRONOGRAMA**

Módulo Regular

TIPO DE PRUEBA	DESCRIPCIÓN NOTA	NÚM. DE PRUEBA	FECHA	OBSERVACIÓN	RECUPERABLE
EA	EVALUACIÓN PARCIAL	1	Semana 8		SÍ
EB	EVALUACIÓN FINAL	1	Semana 15		SÍ
PC	PRÁCTICAS PC	1	Semana 6		SÍ
PC	PRÁCTICAS PC	2	Semana 11		SÍ
TF	TRABAJO FINAL	1	Semana 13		NO

**IX. BIBLIOGRAFÍA DEL CURSO**

**BÁSICA**

Budd, Timothy (1994) Introducción a la programación orientada a objetos / 005.117 BUDD Wilmington, DL : Addison-Wesley Iberoamericana, 1994 de de

Fowler, Martín (1997) UML distilled : applying the standard object modeling language / 005.117 FOWL Reading, MA : Addison-Wesley, 1997 de de

Joyanes Aguilar, Luis (2000) Programación orientada a objetos / 005.117 JOYA Madrid : McGraw-Hill, 2000 de de

**RECOMENDADA**

(No necesariamente disponible en el Centro de Información)

BELL, Douglas (2003) Java para estudiantes. México, D.F. : Pearson Educación.

(005.133J BELL)

- CEBALLOS SIERRA, Francisco Javier (2006) Java 2 : curso de programación. Madrid : Alfaomega.  
(005.133J CEBA/J 2006)
- DEITEL, Harvey (2004) Cómo programar en Java. México, D.F. : Pearson Educación.  
(005.133J DEIT 2004)
- FROUFE QUINTAS, Agustín (2006) Java 2 : manual de usuario y tutorial. México, D.F. : Alfaomega.  
(005.133J2SE FROU)
- HORSTMANN, Cay S. (2005) Core Java 2 : volumen I - fundamentals. Santa Clara California : Sun Microsystems Press.  
(005.133J HORSE)
- KEEGAN, Patrick (2006) NetBeans IDE field guide : developing desktop, Web, enterprise, and mobile applications. Upper Saddle River, NJ : Prentice Hall.  
(005.133J KEEG)