



I. INFORMACIÓN GENERAL

CURSO	:	Programación Orientada a Objetos
CÓDIGO	:	IS210
CICLO	:	201901
CUERPO ACADÉMICO	:	Flores Orihuela, Carlos Alberto Jacinto Gutarra, Jorge Lorenzo Narvaez Villacorta, Jorge Rosvin
CRÉDITOS	:	4
SEMANAS	:	10
HORAS	:	4 H (Laboratorio) Semanal /6 H (Teoría) Semanal
ÁREA O CARRERA	:	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

Descripción: El curso está dirigido a los estudiantes del Ciclo 2 y brinda los conocimientos necesarios al futuro Ingeniero de Sistemas sobre la programación orientada a objetos, conceptos y elementos básicos, abstracción de clases, desarrollo de métodos, aplicación práctica de los conceptos principales de la programación orientada a objetos como son: encapsulamiento, polimorfismo, herencia y uso de bibliotecas de código. Se hace énfasis en el análisis del problema, diseño del diagrama de clases, diseño del juego de datos de prueba, codificación y refinamiento. Estos conceptos permitirán la elaboración de un programa de computadora orientados a objetos que resuelvan tareas simples de forma estandarizada y precisa para la obtención de los resultados esperados.

Propósito: El propósito del curso es que el estudiante desarrolle las habilidades de abstracción de una realidad y la plasme en un programa orientado a objetos. El curso contribuye al desarrollo de la competencia general: Razonamiento Cuantitativo y a la competencia específica Propone soluciones en Ingeniería de Sistemas aplicando principios de matemática, ciencias, computación e ingeniería. El curso, además, tiene como requisito el curso de Fundamentos de Programación y es a su vez requisito de Desarrollo para Entorno Web.

IV. LOGRO (S) DEL CURSO

Al finalizar el curso el estudiante construye programas en la resolución de problemas por computadoras, aplicando fundamentos y principios de la programación orientada a objetos.

-Competencia General:Razonamiento Cuantitativo

-Nivel: 1

-Definición: Capacidad para interpretar, representar, comunicar y utilizar información cuantitativa diversa en situaciones de contexto real. Incluye calcular, razonar, emitir juicios y tomar decisiones con base en esta información cuantitativa.

-Competencia Específica: Propone soluciones en Ingeniería de Sistemas aplicando principios de matemática, ciencias, computación e ingeniería.

-Nivel: 1

-Definición: Soluciona problemas de ingeniería aplicando principios de matemática, ciencias, computación e ingeniería.

V. UNIDADES DE APRENDIZAJE

UNIDAD N°: 1 Clases y Objetos

LOGRO

Al finalizar la unidad, el estudiante aplica los conceptos de clases y objetos de manera práctica en la resolución de problemas simples.

TEMARIO

Contenido (temario):

- Programación Orientada a Objetos: características y ventajas
- Conceptos básicos de Programación Orientada a Objetos: Clases, Objetos, Métodos.
- Propiedades, métodos, clases e instancias

Actividades de aprendizaje:

- Lluvia de ideas inicial sobre el concepto de programación y objeto.
- Cuadro creado por el instructor de los conceptos de programación orientada a objetos.
- Cuadro creado por el profesor del lenguaje de programación Ruby y su sintaxis básica.
- Mapa conceptual de las herramientas de desarrollo.
- Vídeo: Introducción a Ruby.
- Vídeo: Clases e Instancias.
- Participación en el foro solucionando preguntas de compañeros.
- Desarrollos del balotario de ejercicios tipo.

Evidencias de aprendizaje:

- Resolución grupal de un caso propuesto por los alumnos y desarrollar la representación en Objetos considerando la abstracción de los mismos.
- Consultas de los alumnos en el foro y respuesta de los estudiantes a consultas de sus compañeros.
- Actividad 1 (ACT1): Resolución de ejercicios donde se realizará la abstracción para el modelamiento de clases.
- Participación en el foro (PA1). Se realizarán consultas por parte de los estudiantes, los estudiantes, además, compartirán ejercicios resueltos y responderán consultas de otros estudiantes.

Bibliografía:

- BLACK, David A. (2009) The well-grounded Rubyist. Greenwich, Connecticut Manning ; London. (005.133R BLAC)
- FLANAGAN, DavidMatsumoto, Yukihiro. (2008) The Ruby programming language. Beijing ; Sebastopol, California O'Reilly. (005.133R FLAN)

-OLSEN, Russ. (2011) Eloquent Ruby. Upper Saddle River, Nueva Jersey : Addison-Wesley. (005.133R FERN/E)

HORA(S) / SEMANA(S)

Sesión 1-5

UNIDAD N°: 2 RELACIONES ENTRE CLASES

LOGRO

Al finalizar la unidad el estudiante desarrolla programas utilizando las relaciones entre clases de un sistema de manera correcta.

TEMARIO

Contenido (temario)

-Agregación y Composición- Diagrama de Clases- Cardinalidad.- Colecciones, arreglos.

Actividades de aprendizaje

- Lluvia de ideas sobre el concepto de composición y diagrama de clases.
- Cuadro creado por el instructor de los conceptos de relaciones entre clases.
- Resolución de problemas en grupos, aplicando programación orientada a objetos.
- Visualización y reflexión sobre el material de trabajo autónomo de la unidad.
- Se desarrolla el simulacro de la práctica calificada 1

Evidencias de aprendizaje

- Participación (PA2) Se realizarán consultas por parte de los estudiantes, además compartirán ejercicios resueltos y responderán consultas de otros estudiantes.
- Actividad 2(ACT2) Resolución de ejercicios donde se realizará la abstracción para el modelamiento de clases y sus relaciones.

Bibliografía

- BLACK, David A. (2009) The well-grounded Rubyist. Greenwich, Connecticut Manning ; London. (005.133R BLAC)
- FLANAGAN, DavidMatsumoto, Yukihiro. (2008) The Ruby programming language. Beijing ; Sebastopol, California O'Reilly. (005.133R FLAN)
- OLSEN, Russ. (2011) Eloquent Ruby. Upper Saddle River, Nueva Jersey : Addison-Wesley. (005.133R FERN/E)

HORA(S) / SEMANA(S)

Sesión 6-7

UNIDAD N°: 3 HERENCIA Y POLIMORFISMO

LOGRO

Al finalizar la unidad el estudiante desarrolla programas básicos aplicando los principios de herencia y polimorfismo.

TEMARIO

Contenido (temario)

- Encapsulamiento: modificadores de acceso, visibilidad.
- Herencia.
- Polimorfismo.

Actividades de aprendizaje

- Lluvia de ideas sobre el concepto de herencia.
- Cuadro creado por el instructor de los conceptos de herencia. Implementación en Ruby.
- Demostración de los conceptos con ejercicios desarrollados en las herramientas del curso.
- Visualización y reflexión sobre el material autónomo sobre polimorfismo.
- Aplicación de la Práctica Calificada 1
- Participación en el foro.

Evidencias de aprendizaje

- Práctica calificada 1. Los estudiantes desarrollarán de manera individual dos ejercicios y responderán cinco preguntas de teoría.
- Participación (PA3). Se realizarán consultas por parte de los estudiantes, los estudiantes, además, compartirán ejercicios resueltos y responderán consultas de otros estudiantes.
- Actividad 3(ACT3) Resolución de ejercicios donde se realizará la abstracción para el modelamiento de clases y sus relaciones incluyendo la herencia y polimorfismo.

Bibliografía

- BLACK, David A. (2009) The well-grounded Rubyist. Greenwich, Connecticut Manning ; London. (005.133R BLAC)
- FLANAGAN, DavidMatsumoto, Yukihiro. (2008) The Ruby programming language. Beijing ; Sebastopol, California O'Reilly. (005.133R FLAN)

HORA(S) / SEMANA(S)

Sesión 8-9

UNIDAD N°: 4 PATRONES DE DISEÑO

LOGRO

Al finalizar la unidad el estudiante aplica patrones de diseño en la solución de programas informáticos.

TEMARIO

Contenido (temario)

- Concepto de patrón.
- Patrones notables.

Actividades de aprendizaje

- Debate sobre ¿Cómo aseguramos la calidad del software?
- Cuadro elaborado por el instructor de los conceptos de patrones de diseño. Patrones de diseño Singleton y Observer. Implementación en Ruby.
- Participación en el foro.
- Visionado y reflexión del material de trabajo autónomo sobre patrones.
- Se desarrolla el simulacro para la práctica calificada 2
- Lluvia de ideas sobre el patrón MVC.
- Cuadro preparado por el profesor de los conceptos teóricos del patrón MVC.
- Desarrollo de un caso que aplica el patrón MVC.
- Ejercicios grupales sobre casos que apliquen el patrón MVC.

Evidencias de aprendizaje

- Los estudiantes desarrollarán ejercicios de resolución de casos donde aplicarán conceptos de patrones.
- Los estudiantes desarrollarán ejercicios individuales de implementación de los patrones notables.
- Participación (PA4). Se realizarán consultas por parte de los estudiantes, los estudiantes, además, compartirán ejercicios resueltos y responderán consultas de otros estudiantes.
- Actividad 4(ACT4) Resolución de ejercicios donde se realizará la implementación de patrones .
- Práctica Calificada 2. Los estudiantes desarrollarán de manera individual dos ejercicios prácticos, aplicando los conceptos adquiridos.

Bibliografía

- BLACK, David A. (2009) The well-grounded Rubyist. Greenwich, Connecticut Manning ; London. (005.133R BLAC)
- FLANAGAN, DavidMatsumoto, Yukihiro. (2008) The Ruby programming language. Beijing ; Sebastopol, California O'Reilly. (005.133R FLAN)
- FLANAGAN, DavidMatsumoto, Yukihiro. (2008) The Ruby programming language. Beijing ;Sebastopol, California O'Reilly. (005.133R FLAN)
- OLSEN, Russ. (2011) Eloquent Ruby. Upper Saddle River, Nueva Jersey : Addison-Wesley. (005.133R FERN/E)

HORA(S) / SEMANA(S)

Sesión 10-13

UNIDAD N°: 5 EXCEPCIONES

LOGRO

Al finalizar la unidad el estudiante construye programas aplicando fundamentos y principios de la programación orientada a objetos con un adecuado manejo de "excepciones".

TEMARIO

Contenido (temario)

- Excepciones.
- Programación de Excepciones

Actividades de aprendizaje

- Lluvia de ideas del concepto de excepciones.
- Desarrollo de ejercicios tipo que aplican el concepto de excepciones.
- Participación en el foro
- Construcción de un programa completo con todos los conceptos aplicados.

Evidencias de aprendizaje

- Se realizarán consultas por parte de los estudiantes para los Trabajos Finales en FORO.
- Avances y Revisión del Trabajo Final
- Examen Final. El estudiante desarrollará ejercicios que engloban todos los conceptos desarrollados durante el curso.

Bibliografía

- BLACK, David A. (2009) The well-grounded Rubyist. Greenwich, Connecticut Manning ; London. (005.133R BLAC)
- FLANAGAN, DavidMatsumoto, Yukihiro. (2008) The Ruby programming language. Beijing ; Sebastopol, California O'Reilly. (005.133R FLAN)
- OLSEN, Russ. (2011) Eloquent Ruby. Upper Saddle River, Nueva Jersey : Addison-Wesley. (005.133R FERN/E)

VI. METODOLOGÍA

El Modelo Educativo de la UPC asegura una formación integral, que tiene como pilar el desarrollo de competencias, las que se promueven a través de un proceso de enseñanza-aprendizaje donde el estudiante cumple un rol activo en su aprendizaje, construyéndolo a partir de la reflexión crítica, análisis, discusión, evaluación, exposición e interacción con sus pares, y conectándolo con sus experiencias y conocimientos previos. Por ello, cada sesión está diseñada para ofrecer al estudiante diversas maneras de apropiarse y poner en práctica el nuevo conocimiento en contextos reales o simulados, reconociendo la importancia que esto tiene para su éxito profesional.

El curso es de carácter teórico-práctico, se dicta en formato blended y está distribuido en sesiones presenciales y virtuales.

-Durante las sesiones presenciales, el docente induce al estudiante a definir la aplicación de los temas tratados en las secciones en el entorno profesional donde se desarrolla. Esta definición se llevará al plano general mediante lluvia de ideas y debates basados en las experiencias profesionales de los alumnos. De igual manera, el docente, imparte la base teórica y desarrolla ejemplos prácticos sobre la plataforma seleccionada al inicio del curso. Los estudiantes diseñan y desarrollan en equipos actividades durante el ciclo para aplicar lo aprendido, orientados a la aplicación de conocimientos de ciencias. Las actividades son evaluadas incrementalmente durante controles de avance y evaluaciones en las sesiones de clase programadas.

-Durante las sesiones virtuales el estudiante estudia los materiales de autoestudio complementario (materiales de trabajo autónomo y bibliografía recomendada) disponibles en el aula virtual, investiga de forma autónoma temas sobre los contenidos del curso, desarrolla las actividades sugeridas en la Guía del estudiante, rinden evaluaciones de desempeño a través del aula virtual y/o participa de los foros propuestos por el docente como parte de la nota de participación. En dichos foros realizarán consultas y, además, colaborarán con el docente para resolver las consultas de sus compañeros. De igual manera, el profesor creará grupos en redes sociales para compartir conocimiento generado, tanto por él, como por los alumnos y compartidos con todo el grupo.

El estudiante debe dedicarle un mínimo de cuatro horas semanales fuera de clases para consolidar el aprendizaje.

VII. EVALUACIÓN**FÓRMULA**

$$20\% (PC1) + 20\% (PC2) + 5\% (PA1) + 25\% (EB1) + 10\% (TA1) + 20\% (TF1)$$

TIPO DE NOTA	PESO %
PC - PRÁCTICAS PC	20
PC - PRÁCTICAS PC	20
PA - PARTICIPACIÓN	5
TA - TAREAS ACADÉMICAS	10
TF - TRABAJO FINAL	20
EB - EVALUACIÓN FINAL	25

VIII. CRONOGRAMA

Módulo Regular

TIPO DE PRUEBA	DESCRIPCIÓN NOTA	NÚM. DE PRUEBA	FECHA	OBSERVACIÓN	RECUPERABLE
PC	PRÁCTICAS PC	1	Sesión 8	Unidad 1-2 Individual	SÍ
PC	PRÁCTICAS PC	2	Sesión 12	Unidad 1-3 Individual	SÍ
PA	PARTICIPACIÓN	1	Sesión 16	Unidad 1-5 Individual	NO
TA	TAREAS ACADÉMICAS	1	Sesión 16	Unidad 1-5 Individual	NO
TF	TRABAJO FINAL	1	Sesión 16	Unidad 1-5 Individual	NO
EB	EVALUACIÓN FINAL	1	Sesión 17	Unidad 1-5 Individual	SÍ

IX. BIBLIOGRAFÍA DEL CURSO

https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/4378207930003391?institute=51UPC_INST&auth=LOCAL