CURSO	:	**PROGRAMACIÓN AVANZADA**
TRADUCCIÓN	:	ADVANCED COMPUTER PROGRAMMING
SIGLA	:	IIC2233
CRÉDITOS	:	10
MÓDULOS	:	03
REQUISITOS	:	IIC1103 - INTRODUCCIÓN A LA PROGRAMACIÓN
CARÁCTER	:	MÍNIMO
DISCIPLINA	:	INGENIERÍA
HORARIO CÁT:		M-J: 3
HORARIO AYUD:		L : 6
PROFESORES:		SECCIÓN 1: KARIM PICHARA
		SECCIÓN 2: CHRISTIAN PIERINGER
AYUDANTE JEFE:		BELÉN SALDÍAS (bcsaldias@uc.cl)

I. DESCRIPCIÓN

Este curso enseña técnicas para diseñar, implementar, ejecutar y evaluar herramientas de software que resuelven problemas algorítmicos a partir de especificaciones detalladas. En particular, el curso enseña construcciones avanzadas de programación orientada a objetos, estructuras de datos fundamentales, diseño básico de algoritmos y técnicas de análisis.

II. OBJETIVOS

Al finalizar el curso el alumno será capaz de:

1. Descomponer problemas grandes para diseñar y estructurar sus soluciones.
2. Crear diseños orientados a objetos para problemas simples y comunicar estos diseños a través de documentación externa y comentarios en el código.
3. Aplicar conceptos de orientación a objetos (herencia, polimorfismo, interfaces) y estructuras de datos fundamentales (listas ligadas, stacks, colas, árboles binarios y tablas de hash), para diseñar y escribir programas complejos en el lenguaje de programación Python, pudiendo extender este conocimiento a distintos lenguajes.
5. Usar herramientas de programación comunes (debuggers y sistemas de control de versiones); técnicas de programación (bibliotecas de programación orientada a objetos y pruebas unitarias); y un entorno de desarrollo de software para editar, compilar, y depurar programas.
6. Generar software desde cero, con código de alto nivel, de fácil re-utilización, actualización y mantenimiento. Incluyendo interfaces gráficas significativas, totalmente funcionales.

III. CONTENIDOS

Programación Orientada a Objetos: Objetos, Herencia, Polimorfismo, herencia múltiple, propiedades.
Estructuras de Datos: árboles, diccionarios, colas, stacks, sets.
Funciones de Python y Programación Funcional: Algunas funciones especiales de Python, Comprensión de listas, iterables e iteradores, generadores, funciones lambda, Map, Reduce, Filter, decoradores.
Meta Clases: Comprender la lógica detrás de la construcción y creación de clases
Clases Abstractas: Herencia a partir de clases no instanciables
Simulación: Introducción a la simulación y SimPy
Manejo de Excepciones: Tipos de excepciones/errores y cómo controlarlos
Testing: Tests unitarios en pytest
Interfaces Gráficas: Introducción a las interfaces gráficas usando PyQt4
I/O: Bytes, serialización, audio, imagen
Threading: Creación y sincronización de threads, concurrencia
Networking: Sockets, cliente, servidor
Webservices: Expresiones regulares, uso de servicios REST

IV. METODOLOGÍA

Módulos semanales:
- Cátedra: 2
- Ayudantía/Laboratorio 1

El curso se realiza utilizando metodologías de enseñanza teóricas-prácticas centradas en el alumno que permitan a los estudiantes desarrollar las competencias definidas en los objetivos del curso, tanto en cátedras como en ayudantías.

Este curso está diseñado de forma tal que el alumno dedique al estudio personal un promedio de 6 hrs. a la semana.

V. EVALUACIÓN

Las evaluaciones serán por medio de actividades prácticas en clases, controles, tareas, 2 Interrogaciones y un Examen. La nota final del curso (**NF**) se calculará como:

$$\mathbf{NF} = 0.2 * \mathbf{I} + 0.2 * \mathbf{E} + 0.3 * \mathbf{T} + 0.2 * \mathbf{AC} + 0.1 * \mathbf{C}$$

Donde **I** es el promedio de las interrogaciones, **E** es la nota del Examen, **T** es el promedio de las tareas, **AC** es el promedio de las actividades en clases y **C** es el promedio de los controles. Durante el semestre **NO** se borrará ninguna evaluación, tampoco existe la posibilidad de ser eximido del Examen final.

VI. BILBIOGRAFÍA

Textos Complementarios

- Beazley, David M; Jones, Brian K. — Python Cookbook, 3rd Edition O'Reilly, 2013
- Harrison, Matt — Intermediate Python Programming: Learn Decorators, Generatorsm Functional Programming and More, Kindle Edition, 2013
- Lutz, Mark — Learning Python: Powerful Object-Oriented Programming, 4th Edition O'Reilly, 2009
- Lutz, Mark — Programming Python, 4th Edition O'Reilly, 2010
- Phillips, Dusty — Python 3 Object Oriented Programming, 1st Edition Packt Publishing Ltd., 2010
- Summerfield, Mark — Python in Practice: Create Better Programs Using Concurrency, Libraries, and Patterns, 1st Edition Developer's Library, 2013

VII. POLÍTICA DE INTEGRIDAD ACADÉMICA DEL DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

Los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que **mantengan altos estándares de honestidad académica**. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería (disponible en SIDING).

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno, sin apoyo en material de terceros. Por "trabajo" se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.

En particular, si un alumno copia un trabajo, o si a un alumno se le prueba que compró o intentó comprar un trabajo, obtendrá nota final 1.1 en el curso y se solicitará a la Dirección de Docencia de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral.

Por "copia" se entiende incluir en el trabajo presentado como propio, partes hechas por otra persona. **En caso que corresponda a "copia" a otros alumnos, la sanción anterior se aplicará a todos los involucrados**. En todos los casos, se informará a la Dirección de Docencia de la Escuela de Ingeniería para que tome sanciones adicionales si lo estima conveniente.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la referencia correspondiente.

Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile*. Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.

* Reglamento del Alumno dela Pontificia Universidad Católica de Chile disponible en: http://admisionyregistros.uc.cl/alumnos/informacion-academica/reglamentos-estudiantiles