

IIC2233

PROGRAMACIÓN AVANZADA

Departamento de Ciencia de la Computación

Escuela de Ingeniería

Pontificia Universidad Católica de Chile



Programa

Sigla:	IIC2233
Nombre del curso:	Programación Avanzada
Créditos:	10
Módulos docentes:	Martes y Jueves (2):10:00 – 11:20 hrs.
Profesor:	Karim Pichara Baksai Christian Pieringer Baeza
Ayudantes Jefe:	Belén Saldías Patricio López Jaime Castro Bastián Mavrakis Marco Bucchi
Semestre:	2015-2

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.

OBJETIVOS

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.
4. 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.
5. Generar software desde cero, con código de alto nivel, de fácil re-utilización, actualización y mantenimiento. Incluyendo interfaces gráficas significativas, totalmente funcionales.

CONTENIDO	
1. Programación Orientada a Objetos	1.1 Objetos 1.2 Herencia 1.3 Herencia múltiple 1.4 Propiedades
2. Estructuras de Datos	2.1 Árboles 2.2 Diccionarios 2.3 Colas 2.4 Stacks 2.5 Sets
3. Funciones de Python y Programación Funcional	3.1 Algunas funciones especiales de Python 3.2 Comprensión de listas 3.3 Iterables e iteradores 3.4 Generadores 3.5 Funciones lambda 3.6 Map 3.7 Reduce 3.8 Filter 3.9 Decoradores
4. Meta Clases	4.1 Comprender la lógica detrás de la construcción y creación de clases
5. Clases Abstractas	5.1 Herencia a partir de clases no instanciables
6. Manejo de Excepciones	6.1 Tipos de excepciones/errores 6.2 Controlarlos de excepciones
7. Testing	7.1 Test unitarios en Pytest 7.2 Test unitarios en Unittest
8. Threading	8.1 Creación y sincronización de threads 8.2 Concurrencia
9. Simulación	9.1 Introducción a la simulación con Eventos Discretos 9.2 SimPy 3
10. Interfaces Gráficas (GUI)	10.1 Introducción a las Interfaces gráficas usando PyQt4
11. I/O	11.1 Bytes 11.2 Serialización 11.3 Audio 11.4 Imagen
12. Networking	12.1 Socket, cliente, servidor
13. WebServices	13.1 Expresiones regulares 13.2 Uso de servicios REST

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.

EVALUACIÓN

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

$$NF = 0.2 \times I + 0.2 \times E + 0.3 \times T + 0.2 \times AC + 0.1 \times C$$

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.
- Las tareas son reprobatorias. Si la nota en las tareas, $T < 4.0$, $NF = NT$.
- Solo se aproximará la nota final NF. Todo el resto de las notas serán usadas con dos decimales.
- La inasistencia a una de las interrogaciones (I) podrá ser reemplazada por la nota del examen. Solo se podrá faltar a una de las interrogaciones y esta deberá ser justificada con autorización de la Dirección de Pregrado. No se puede faltar al examen.

FECHAS INTERROGACIONES

- Interrogación 1: lunes 21 de Septiembre, 18:30 hrs.
- Interrogación 2: viernes 13 de Noviembre, 18:30 hrs.
- Examen: jueves 3 de Diciembre

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:

http://wwwold.ing.puc.cl/esp/alumpregrado/procedimientos/integridad_academica.html

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 (uno punto uno) 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 otro alumno. En este caso, la sanción anterior se aplicará a todos los alumnos involucrados. Por "compra" se entiende presentar como propio un trabajo hecho por otra persona. En este caso, se informará a la Dirección de Docencia la persona que vende el trabajo.

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:

<http://dsrd.uc.cl/alumnos-uc/reglamento-estudiantiles>

Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.