

IIC2233

## PROGRAMACIÓN AVANZADA

Departamento de Ciencia de la Computación

Escuela de Ingeniería

Pontificia Universidad Católica de Chile



## Programa

<b>Sigla:</b>	IIC2233
<b>Nombre del curso:</b>	Programación Avanzada
<b>Créditos:</b>	10
<b>Módulos docentes:</b>	Jueves (4, 5): 14:00 – 17:00 hrs. Lunes (6): 17:00 hrs.
<b>Profesores:</b>	Karim Pichara Baksai Christian Pieringer Baeza
<b>Ayudantes Jefe:</b>	Coordinación: Jaime Castro Docencia: Ivania Donoso, Antonio Ossa Tareas: Bastián Mavrakis, Matías Jünemann Supervisor: Belén Saldías
<b>Semestre:</b>	2016-1

### 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 reutilización, 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 Clases abstractas 1.5 Properties
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. Manejo de Excepciones	5.1 Tipos de excepciones 5.2 Control de excepciones
6. Testing	6.1 Tests unitarios en Unittest
7. Simulación	7.1 Introducción a la simulación DES (Discrete Event Simulation)
8. Threading	8.1 Creación y sincronización de threads 8.2 Concurrencia
9. Interfaces Gráficas (GUI)	9.1 Introducción a las Interfaces gráficas usando PyQt4
10. I/O	10.1 Manejo de Strings 10.2 Bytes 10.3 Serialización en formato binario 10.4 Serialización en formato JSON
11. Networking	11.1 Sockets, cliente, servidor

## 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 7 hrs. a la semana.

## EVALUACIÓN

- Las evaluaciones serán efectuadas por medio de actividades prácticas en clases, controles (de materia y/o lecturas adicionales), tareas, un midterm test y un examen final. Se calculará la nota del curso (NC) como

$$NC = 0.15 \times MT + 0.2 \times FE + 0.3 \times T + 0.25 \times AC + 0.1 \times C$$

Donde MT es la nota obtenida en el midterm test, FE es la nota del examen final, T es el promedio de las tareas, AC es el promedio de las actividades en clases sin incluir la peor de ellas, y C es el promedio de los controles.

- Adicionalmente, para aprobar el curso el alumno debe cumplir con:
  - o El promedio ponderado entre MT y FE debe ser mayor o igual a 3.500
  - o AC debe ser mayor o igual a 3.700
  - o T debe ser mayor o igual a 3.950
- Si el alumno cumple con todas las condiciones mencionadas, la nota final del curso (NF) corresponde a NC. En caso contrario, NF es la nota mínimo entre los ítems no cumplidos.
- Durante el semestre **NO** se borrará ninguna evaluación. Tampoco existe la posibilidad de ser eximido del Examen final.
  - o Sin embargo, **se tomará el promedio AC sin la peor actividad.**
  - o Además **existe la posibilidad de que se anuncie una tarea recuperativa opcional<sup>1</sup>**. Si ésta es realizada por el alumno, tiene la posibilidad de eliminar la peor de las tareas bajo ciertas condiciones que serán comunicadas oportunamente.
- Solo se aproximará la nota final NF. Todo el resto de las notas serán usadas con dos decimales.
- No se podrá faltar al midterm o el examen final. **La inasistencia a cualquiera de ellos es reprobatoria.**

## FECHAS INTERROGACIONES

- Midterm test: 17 de mayo 18:30
- Final test: 5 de julio 09:15

---

<sup>1</sup> Se anunciará su realización o no realización antes del 1 de junio de 2016.

## INTEGRIDAD EN LAS EVALUACIONES

Desacatar las indicaciones de cada evaluación tiene como sanción inmediata un 1.0 en dicha evaluación. Por ejemplo: Cambios arbitrarios en las parejas de trabajo, no respetar los medios de entrega de evaluaciones, formatos, etc.

Cualquier situación de copia en alguna evaluación tendrá como sanción un 1.1. final en el curso. Esto sin perjuicio de sanciones posteriores que estén de acuerdo a la Política de Integridad Académica del Departamento de Ciencia de la Computación, la Escuela de Ingeniería y de la Universidad, que sean aplicables para el caso.

### 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](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.