Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2115 - PROGRAMACIÓN COMO HERRAMIENTA PARA LA INGENIERÍA

– Programa de curso –

Profesor	:	Hans Löbel (halobel@ing.puc.cl)
Sitio Web	:	github.com/IIC2115/Syllabus
Clases	:	jueves, módulos 4 y 5 (14:00 - 17:00) - Sala K201
Ayudantías	:	martes, módulo 4 (14:00 - 15:20) - Sala E10
Horario de atención	:	agendar cita por email

1 Presentación del curso

Durante los últimos años, el uso y desarrollo de software especializado en las distintas especialidades de la ingeniería se ha transformado en una constante, ya sea por lo complejo de las tareas a realizar, o por la gran cantidad de datos que es necesario analizar. Es por esto que el conocimiento y las habilidades relacionadas con la programación se han transformado no sólo en una ventaja, sino en una necesidad para los profesionales de la ingeniería.

El propósito de este curso es que el alumno se familiarice con la programación como una básica y poderosa herramienta, no sólo para solucionar de manera más eficiente y efectiva problemas clásicos en ingeniería, sino que además para desarrollar soluciones innovadoras a nuevos problemas. Para alcanzar este objetivo, el curso cubre una amplia variedad de lenguajes y herramientas que son (y serán) fundamentales para enfrentar de manera satisfactoria problemas de ingeniería, tanto en el aspecto profesional, como en el académico.

2 Objetivos de aprendizaje

A nivel general, al finalizar el curso los alumnos serán capaces de:

- Evaluar y utilizar de manera efectiva distintos lenguajes y herramientas de programación para resolver problemas asociados a sus áreas de especialización, en base a los requerimientos de estos.
- Proponer y desarrollar soluciones novedosas utilizando la programación, ya sea para problemas tradicionales o para nuevos problemas en ingeniería.

A nivel particular, al finalizar el curso los alumnos serán capaces de:

- Utilizar herramientas modernas para el desarrollo de software.

- Modelar un problema utilizando técnicas de programación orientada a objetos.
- Crear soluciones a problemas utilizando estructuras y técnicas avanzadas de programación.
- Modelar datos y sus relaciones, y realizar consultas sobre estos, mediante distintos modelos y lenguajes.
- Analizar, visualizar y presentar datos utilizando distintos lenguajes.
- Formatear y confeccionar documentos técnicos mediante herramientas especializadas.

3 Contenido

A continuación se presenta un desglose detallado de los contenidos del curso. El orden en que estos se cubrirán no tendrá necesariamente que ver con el orden en que estos aparecen en la siguiente lista:

Introducción y herramientas básicas

- Tipos de lenguaje de programación.
- Interpretación y compilación.
- Sistemas de control de versiones (git).

Desarrollo avanzado en Python

- Programación orientada a objetos.
- Estructuras de datos.
- I/O.
- Simulación

Técnicas para resolución de problemas

- Recursión.
- Dividir y conquistar.
- Backtracking.
- Ordenamiento y búsqueda.
- Búsqueda en grafos.

Uso de bases de datos

- Modelo relacional de datos.
- Consultas sobre datos usando SQL.

Tópicos avanzados

- Confección de documentos con LaTeX.
- Análisis y visualización de datos con Python.

4 Metodología

El curso sigue una metodología de clase invertida (*flipped classroom*), donde los alumnos deben estudiar los contenidos de manera previa a la clase, para luego aplicarlos en ella mediante actividades prácticas de programación. El material de estudio se encontrará disponible en el sitio del curso y consiste en apuntes, manuales y libros donde se describen detalladamente los tópicos. Se espera además que los alumnos utilicen otras fuentes para complementar y profundizar los contenidos.

Las sesiones de cátedra consisten en un breve repaso inicial de los tópicos a cubrir en la sesión (15-20 minutos), para posteriormente dedicar el resto del tiempo a la actividad práctica correspondiente. Las actividades se realizarán de manera individual o en grupos de dos, con la asistencia del profesor y de los ayudantes, quienes estarán disponibles para contestar dudas y aclarar conceptos. La asistencia a clases es "voluntaria", pero es importante considerar que aproximadamente la mitad de las actividades prácticas serán evaluadas, por lo que la inasistencia a estas resultará en una calificación de 1.0 en la actividad.

En cinco de las sesiones de cátedra se realizarán evaluaciones en forma de controles cortos, que cubrirán tanto la materia de los apuntes como el contenido de las actividades prácticas.

Finalmente, como apoyo complementario a las evaluaciones del curso, se realizarán ayudantías para cada una de las tareas, donde se explicará en más detalle el enunciado y se darán ejemplos y consejos para la realización de esta.

5 Evaluaciones

Las evaluaciones se dividen en cuatro tipos, cada una con su correspondiente nota final promedio:

- Actividades prácticas (30%): cada dos semanas, las actividades prácticas en clases serán evaluadas, en base a su completitud y la aplicación de los contenidos considerados para esa sesión y la anterior. Las actividades prácticas serán individuales o en grupos de a dos (formados de manera aleatoria), dependiendo de los contenidos a cubrir. La nota final de las actividades prácticas (**A**) está dada por el promedio de estas.

- Tareas (40%): se realizarán seis tareas individuales, todas de igual valor, sobre tópicos cubiertos en el material y actividades del curso. Las tareas mezclarán contenido de distintos capítulos y deberán ser entregadas electrónicamente a través de GitHub. Se considerará un descuento por atraso de 1.0 ptos. cada 6 horas o fracción. Las fechas tentativas de las tareas se presentan a continuación, donde la fecha entre paréntesis indica la fecha de entrega:

  - T_1: 29/08 (14/09)
  - T_2: 20/09 (09/10)
  - T_3: 13/10 (29/10)
  - T_4: 15/11 (29/11)

  La nota final promedio de las tareas (**T**) se obtendrá borrando la peor nota obtenida y promediando las cinco restantes.

- Evaluaciones escritas (20%): se realizarán 5 evaluaciones escritas individuales a lo largo del semestre: 4 controles escritos, que evaluarán los tópicos cubiertos en el material del curso y las actividades, y un examen final que evaluará los tópicos de todo el curso. La nota final de las evaluaciones escritas (**E**) está dada por la siguiente fórmula (examen vale por dos, se borra la peor nota):

$$E = \frac{(C_1 + C_2 + C_3 + C_4 + 2 \times Examen) - min(C_1, C_2, C_3, C4, Examen)}{5}$$

- Participación (10%): las sesiones de clases que no consideren actividades prácticas evaluadas, tendrán incidencia en la nota del curso por concepto de asistencia/participación. La nota final de participación (**P**) se calculará usando el porcentaje de sesiones con actividades prácticas no evaluadas en la que se asistió y participó, utilizando una escala lineal.

6 Cronograma de actividades

Fecha	Actividades	Tópicos
03/08	Presentación del curso, Actividad de diagnóstico	Introducción a la programación
10/08	Actividad práctica no evaluada	Orientación a objetos parte I
17/08	Actividad práctica evaluada	Orientación a objetos parte II
24/08	Control 1, Actividad práctica no evaluada	Estructuras de datos parte I
31/08	Actividad práctica evaluada	Estructura de datos parte II
07/09	Actividad práctica no evaluada	Técnicas de programación parte I
14/09	Actividad práctica evaluada	Técnicas de programación parte II
21/09	Control 2, Actividad práctica evaluada	Técnicas de programación parte II
28/09	Actividad práctica no evaluada	I/O
05/10	Ayudantía	I/O
12/10	Actividad práctica no evaluada	Bases de datos parte I
19/10	Actividad práctica evaluada	Bases de datos parte II
26/10	Control 3, Actividad práctica no evaluada	Simulación parte I
02/11	Actividad práctica evaluada	Simulación parte II
09/11	Control 4, Actividad práctica no evaluada	Análisis y visualización de datos parte I
16/11	Actividad práctica evaluada	Análisis y visualización de datos parte II

7 Exigencias del curso

Para aprobar el curso se deben cumplir con los siguiente requisitos:

- Las notas **A**, **T** y **E** deben ser mayores o iguales a 3.95.
- La inasistencia a una evaluación escrita (justificada o no) genera automáticamente nota 1.0 en esta.
- El examen tiene carácter obligatorio y no existe eximición de él.
- En caso de cumplir todos los criterios, la nota final del curso (**F**) se calcula de la siguiente manera:

$$\mathbf{F} = 0.3 \cdot \mathbf{A} + 0.4 \cdot \mathbf{T} + 0.2 \cdot \mathbf{E} + 0.1 \cdot \mathbf{P}$$

  En caso contrario, la nota final de reprobación ($\tilde{\mathbf{F}}$) será:

$$\tilde{\mathbf{F}} = min(3.9, \mathbf{F})$$

8 Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

"Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes 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. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por "copia" o "plagio" se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.

9 Bibliografía

- Apuntes del curso disponibles en el sitio.
- *Advanced Computer Programming in Python*, Pichara, K.; Pieringer, C., 2017.
- *Database Management Systems*, Ramakrishnan, R.; Gehrke, J., 2002.
- *LaTeX Beginner's Guide*, Kottwitz, S., 2011
- *A Smarter Way to Learn JavaScript. The new tech-assisted approach that requires half the effort*, Myers, M., 2014
- *The Art of R Programming: A Tour of Statistical Software Design*, Matloff, N., 2011.