PLANIFICACIÓN DEL CURSO

I. ACTIVIDAD CURRICULAR Y CARGA HORARIA

Asignatura:	Programación	Código: ING1302
Semestre de la Carrera:	I semestre	
Carrera:	Obligatorio para todas las carreras de Ingeniería Civil	
Escuela:	Escuela de Ingeniería	
Docente(s):	Carol Moraga - Raúl Valenzuela	
Ayudante(s):	Por definir	
Horario:	Miércoles 14:30-16:00, Jueves 12:00-13:30, Ayudantía Viernes 8:30-10:00	

Créditos SCT:	6
Carga horaria semestral[1]:	180 horas
Carga horaria semanal:	10 horas

Tiempo de trabajo sincrónico semanal:	3 horas
Tiempo de trabajo asincrónico semanal:	7 horas

II. RESULTADOS U OBJETIVOS DE APRENDIZAJE ESPERADOS ESTE SEMESTRE

1)	Descomponer un problema y hacer abstracciones utilizando el razonamiento lógico y algorítmico.
2)	Plantear la solución a los problemas resultantes de la descomposición: diseñar contratos, especificar el propósito del código, generar casos de prueba y programar la solución.
3)	Detectar y corregir errores de programación.

[1] *Considere que 1 crédito SCT equivale a 27 horas de trabajo total (presencial/sincrónico y autónomo/asincrónico) en el semestre.*

III. UNIDADES, CONTENIDOS Y ACTIVIDADES

UNIDAD 1:			
Semana	**Contenidos**	**Ayudantía**	**Actividades**
1	Introducción/Hardware y lógica en la programación	No hay ayudantía	
2	Estructuras básicas y diagramas de flujo/Ejercicios de estructuración	Feriado	
3	Variables de tipo numérico y string/Input y Bucles (While, For)	Ejercicios de variables y Bucles	Tarea I (Publicación)
4	Introducción a plataformas de ejecución de código/Repaso	Repaso y ejercicios	

UNIDAD 2:			
Semana	**Contenidos**	**Ayudantía**	**Actividades**
5	Sentencia condicional, operador AND y OR/Listas, tuplas y diccionarios	Ejercicios de condición aplicando listas y tuplas	
6	Librería Turtle/Ejercicios con Turtle	Ejercicios con Turtle	Tarea I (Entrega)
7	Feriado/Repaso	Ejercicios de preparación CC1	

8	Funciones I/Funciones II	Ejercicios de funciones	CC1/ Tarea II (Publicación)

UNIDAD 3:			
Semana	**Contenidos**	**Ayudantía**	**Actividades**
9	Recursividad I/Recursividad II	Ejercicios de Recursividad	
10	I/O de archivos/ I/O de archivos	Ejercicios de Archivos	Tarea II (Entrega)
11	Introducción a POO/ Técnicas en POO	Ejercicios de POO	
12	Herencia y polimorfismo/ Repaso	Ejercicios de preparación CC2	Tarea III (Publicación)
13	Funciones y detección de errores/ Feriado	Feriado	CC2
14	Sobrecarga de operadores / Ejercicios avanzados	Ejercicios de preparación Exámen	Tarea III (Entrega)

IV. CONDICIONES Y POLÍTICAS DE EVALUACIÓN

Se evaluará el aprendizaje del contenido presentado, mediante:

- Control de Cátedra 1 (35%)
- Control de Cátedra 2 (35%)
- Actividades Complementarias (30%)

Las actividades complementarias consisten en 3 Tareas, desarrolladas de manera individual. Las Tareas consistirán en la resolución de un problema utilizando lo aprendido en las Cátedras.
El curso se aprueba con un promedio final ponderado igual o mayor a 4,0. Tanto el promedio de los Controles de Cátedra como de las Actividades Complementarias (cada una por separado) deber ser igual o mayor a 4,0 para aprobar el curso. Se eximirá de examen con nota igual o superior a 5,5.

Estudiantes que se ausenten justificadamente a alguno de los Controles de Cátedra tendrán la oportunidad de recuperarlo a través de un control recuperativo (justificativo se tramita a través de Dirección de Asuntos Estudiantiles). Las notas de tareas no se recuperan.

El Examen recuperativo se puede rendir al tener Nota de Cátedra (Controles de Cátedra + Examen) entre 3.7-3.9. Si aprueban el Examen, la Nota de Cátedra es igual a 4.0. Este criterio no se puede modificar sin previa autorización de la Dirección de Escuela.

Un/a estudiante que cometa plagio obtendrá un **1,0** en la evaluación y el caso será informado a Escuela de Ingeniería.

V. BIBLIOGRAFÍA Y RECURSOS OBLIGATORIOS

- Dawson, Michael 2010:. **Python Programming for the Absolute Beginner.** CENGAGE Learning.
- Kernighan, B. and R. Pike, 1999: **The Practice of Programming**. Lucent Technologies.
- Felleisen M., R. Findler, M. Flatt, and S. Krishnamurthi, 2001: **How to Design Programms: an introduction to programming and computing**. MIT Press.
- **Python Tutorial w3schools.com.** https://www.w3schools.com/python/.
- **Farrell, Programing Logic and Design.**

VI. BIBLIOGRAFÍA Y RECURSOS COMPLEMENTARIOS

- Google Colab: https://colab.research.google.com/
- Repositorio en github: https://github.com/rvalenzuelar