FACULTAD DE INGENIERÍA DE SISTEMAS,
INFORMÁTICAY CIENCIAS DE LA COMPUTACIÓN
CIENCIAS DE LA COMPUTACION IV: COMPILADORES

Catedrático Ing. Luis Jimenez luiscu@galileo.edu

Objetivo del curso

Este es un curso introductorio al diseño de lenguajes de programación y la implementación de sus compiladores e intérpretes. En el proceso, haremos una exploración general al diseño de lenguajes de programación y su impacto en la implementación de estos, este semestre, nos enfocaremos en un lenguaje particular: Classroom Object Oriented Language (COOL).

Uno de los objetivos de este curso es explorar la estructura de los lenguajes de programación y considerar alternativas a características familiares en lenguajes de programación. Estudiaremos además el problema de la traducción de lenguajes de alto nivel a formas que pueda ejecutar la máquina, utilizando el lenguaje COOL como lenguaje de alto nivel y RISC-V como lenguaje ensamblador. La implementación de un compilador se hace por varias razones; Primero para aprender técnicas que se aplican a muchas clases de problemas fuera del ámbito de un compilador, segundo, para obtener una mejor comprensión de las herramientas que utilizamos cuando programamos y los costos de los programas que escribimos, y tercero (y probablemente la más importante), para obtener experiencia construyendo y validando un problema de ingeniería sustancialmente complejo.

Competencias

Al finalizar el curso el estudiante tendrá las siguientes competencias:

- Entiende conceptos del diseño de lenguajes de programación y la implementación de compiladores e intérpretes para estos.
- Utiliza herramientas de desarrollo para generar scanners y parsers para resolver distintos problemas y analizar lenguajes de programación.
- Utiliza reglas formales para describir el significado de las estructuras de programación asociadas a un lenguaje, y las acciones que estas representan al momento de la ejecución de un programa.
- Es capaz de traducir programas escritos en lenguajes de alto nivel orientados a objetos, a lenguaje ensamblador.

Prerrequisitos

Es muy importante que tengan claros sus conceptos de sus cursos de Ciencias de la Computación I, II y III, así como de Informática III, dado que este es uno de los cursos que requiere mayor esfuerzo en el área de Ciencias de la Computación. Específicamente estaremos desarrollando el compilador en Java.

Horario del Curso

El curso consta de siete créditos académicos (CA) distribuidos en el siguiente horario:

- Clases magistrales (cuatro períodos a la semana)
- Laboratorios (dos períodos a la semana)
- Discusión y resolución de dudas (un período a la semana)

Evaluación del Curso

Exámenes	30 puntos
Tareas	5 puntos
Proyectos	50 puntos
Laboratorios	15 puntos
Nota Final	100 puntos

El proyecto consta de cinco entregas, la ponderación tentativa es de 15%, 15%, 30%, 30%, 10%.

Requisitos adicionales de aprobación
Para aprobar el curso, el estudiante debe de cumplir con los siguientes requisitos adicionales:

- Asistencia mínima a clase: 80%
- Asistencia mínima a laboratorio: 80%
- Entrega de **todas** las fases del proyecto

Entregas tarde / Casos especiales:
Utilizamos un sistema de redundancia y puntos extras con el objetivo que el estudiante que falta a un par de tareas/exámenes aun pueda obtener la nota de 100 puntos en la clase, sin embargo, si el estudiante completa todas las tareas y nunca falta, podrá obtener una nota superior a 100 puntos (por cuestiones administrativas la nota máxima en expediente es de 100 puntos).

Además, implementamos un sistema de "Slip Days" con el cual el estudiante tiene tres días de entrega tarde en cualquier proyecto o tarea. El estudiante decide en que proyecto o tarea utilizar sus tres días de entrega tardía, excepto los que se indiquen.

Bibliografía
Para este curso no se requiere ningún libro, dado que utilizaremos principalmente las copias de clase y referencias en internet. Algunas personas prefieren adquirir libros de texto, para lo cual recomendamos los siguientes:

Modern Compiler Implementation in Java de Andrew W. Appel.
Compilers: Principles, Techniques and Tools de Aho, Sethi y Ullman

Proyecto
El proyecto **es el curso**. Se llevará a cabo en cinco fases, para los segmentos que requiere un compilador que son: scanner, parser, chequeo semántico, generación de código, y optimizador de instrucciones. Para cada proyecto se les entregarán instrucciones con información más detallada.

Política de copia
Todas las copias son reportadas al director de carrera y al decano de la facultad. A discreción de ellos, se aplicarán las sanciones descritas en el artículo 70 del reglamento de la Universidad.

Si la copia es en una fase del proyecto, tendrá una nota de cero puntos en el proyecto completo por lo cuál no podrá aprobar el curso.

Índice temático
- Course Overview
- Cool: The Course Project
- Lexical Analysis
- Finite Automata
- Parsing
- Top-Down Parsing
- Bottom-Up Parsing I
- Bottom-Up Parsing II
- Semantic Analysis and Type Checking
- Cool Type Checking
- Runtime Organization
- Code Generation
- Operational Semantics
- Local Optimization
- Global Optimization
- Register Allocation
- Garbage Collection
- Java