GUÍA DOCENTE ABREVIADA DE LA ASIGNATURA

G1662 - Lenguajes de Programación

Grado en Ingeniería Informática

Curso Académico 2023-2024

1. DATOS IDENTIFICATIVOS					
Título/s	Grado en Ingeniería Informática			Tipología v Curso	Optativa. Curso 4
Centro	Facultad de Ciencias				
Módulo / materia	MATERIA DE COMPUTACIÓN MENCIÓN EN COMPUTACIÓN				
Código y denominación	G1662 - Lenguajes de Programación				
Créditos ECTS	6	Cuatrimestre		Cuatrimestral (2)	
Web	https://moodle.unican.es/course/view.php?id=12160				
Idioma de impartición	Español	English friendly	No	Forma de impartición	Presencial

Departamento	DPTO. MATEMATICAS, ESTADISTICA Y COMPUTACION
Profesor responsable	DOMINGO GOMEZ PEREZ
E-mail	domingo.gomez@unican.es
Número despacho	Facultad de Ciencias. Planta: + 3. DESPACHO DOMINGO GOMEZ PEREZ (3005)
Otros profesores	

3.1 RESULTADOS DE APRENDIZAJE
- Conocer la aplicación del análisis sintáctico y semántico a la generación de código, comprendiendo a fondo la estructura de un compilador y conociendo la problemática asociada a cada una de sus fases.
- Conocer y saber implementar en fragmentos de compiladores las características de los lenguajes de programación existentes: reglas de "scoping", máquinas virtuales, programación declarativa.
- Saber aplicar estos fundamentos, conjuntamente con el uso de herramientas software, para la construcción de analizadores de lenguajes de programación o de otros lenguajes de especificación, atendiendo correctamente a los problemas de ambigüedad sintáctica y gestión de tablas de símbolos con mecanismos de "scoping"

4. OBJETIVOS
Comprender una de las muchas utilidades de los lenguajes formales
Entender las principales dificultades a la hora de generar código
Utilizar el modelo de autómatas con pila para desarrollar compiladores
Conocer métodos para el tratamiento de errores en código fuente
Utilizar varias herramientas informáticas para generar compiladores

6. ORGANIZACIÓN DOCENTE	
CONTENIDOS	
1	BLOQUE 1: Conceptos teoricos
1.1	Introducción a la asignatura y repaso de lenguajes formales
1.2	Analizadores Ascendentes
1.3	Tipos de analizadores ascendentes: LR(0), SLR, LR(1) y LALR
1.4	Gramáticas de atributos
1.5	Chequeo de tipos y generación de código
1.6	Optimización de código y registros
2	BLOQUE 2: Herramientas
2.1	Python
2.2	Generadores de analizadores léxicos
2.3	Generadores de analizadores sintáticos
2.4	Generación de código
2.5	Traducción a código máquina

7. MÉTODOS DE LA EVALUACIÓN

Descripción	Tipología	Eval. Final	Recuper.	%
Práctica individual	Trabajo	No	Sí	30,00
Prácticas en grupo	Trabajo	No	Sí	30,00
Problemas	Actividad de evaluación con soporte virtual	No	Sí	10,00
Examen Escrito	Examen escrito	Sí	Sí	30,00
TOTAL				100,00

Observaciones

La evaluación continua de problemas, práctica individual y prácticas en grupo se podrá superar en algunos casos mediante un examen de prácticas (70%):
• En el periodo ordinario, para los alumnos que por motivo justificado (estudiantes a tiempo parcial) no hayan seguido la evaluación continua.
• En el periodo de recuperación, para los alumnos que no hayan superado el examen o que quieran mejorar la nota de prácticas.
Será un examen escrito, de dos horas de duración, en el que habrá que resolver varios cuestiones y problemas. Para poder presentarse a este examen se deberán entregar las prácticas en grupo.

En el caso de que las circunstancias sociosanitarias exijan el cambio a una modalidad de teletrabajo, el examen final se sustituirá por un trabajo práctico y un cuestionario teórico en función de los recursos disponibles para su realización, favoreciendose la modalidad asíncrona y no presencial.

Criterios de evaluación para estudiantes a tiempo parcial

Los alumnos que justifiquen estar a tiempo parcial, podrán optar al mismo sistema de evaluación o presentarse al examen final y al examen de recuperación de problemas, práctica individual y prácticas en grupo. En este último caso, la nota final es el resultado de tomar el 50% de la nota del examen final más 50% del examen de recuperación.

8. BIBLIOGRAFÍA Y MATERIALES DIDÁCTICOS

BÁSICA

Torben Mogensen. Basics of compiler design

Alfred Aho, Monica Lam, Ravi Sethi, and Jeffrey Ullman.
Compilers: Principles, Techniques, and Tools (Second Edition)

Esta es la Guía Docente abreviada de la asignatura. Tienes también publicada en la Web la información más detallada de la asignatura en la Guía Docente Completa.