



UNIVERSIDADE FEDERAL DE OURO PRETO  
DEPARTAMENTO DE COMPUTAÇÃO



PLANO DE ENSINO

<b>Nome do Componente Curricular em português:</b> Construção de Compiladores I		<b>Código:</b> BCC328
<b>Nome do Componente Curricular em inglês:</b> Compiler Construction I		
<b>Nome e sigla do departamento:</b> Departamento de Computação (DECOM)		<b>Unidade acadêmica:</b> ICEB
<b>Nome do docente:</b> José Romildo Malaquias		
<b>Carga horária semestral:</b> 60 horas	<b>Carga horária semanal teórica:</b> 8 horas/aula	<b>Carga horária semanal prática:</b> 0 horas/aula
<b>Data de aprovação na assembleia departamental:</b> ___ / ___ / _____		
<b>Ementa:</b> Implementação de linguagens de programação: compilação e interpretação; análise léxica; análise sintática; análise semântica.		
<b>Conteúdo Programático:</b> <ul style="list-style-type: none"><li>• Implementação de linguagens de programação: compilação e interpretação</li><li>• Análise léxica: análise léxica, analisador léxico ad hoc, expressões regulares, autômatos finitos e geradores de analisadores léxicos</li><li>• Análise sintática: análise sintática, gramáticas livre de contexto, análise sintática descendente, análise sintática ascendente, geradores de analisadores sintáticos e ações semânticas</li><li>• Análise semântica: árvore abstrata, tabelas de símbolos, checagem de tipos e regras semânticas</li></ul>		
<b>Objetivos:</b> <p>Ao final do curso é esperado que o aluno:</p> <ul style="list-style-type: none"><li>- possua uma visão geral sobre o processo de compilação sob o ponto de vista de implementação</li><li>- saiba especificar aspectos léxicos, sintáticos e semânticos de linguagens através de expressões regulares e gramáticas livres de contexto</li><li>- conheça as principais técnicas e ferramentas de apoio usadas na construção de compiladores, sabendo usá-las na especificação e implementação de linguagens de programação</li></ul>		
<b>Metodologia:</b> <p><b>Aulas expositivas</b> usando vídeos online (possivelmente ao vivo) para apresentação do conteúdo, e disponibilizados para os alunos.</p> <p><b>Exercícios de fixação</b> para explorar a base teórica do conteúdo programático.</p> <p><b>Exercícios práticos</b> para aplicação de técnicas de compilação na construção do <i>front end</i> de um compilador para uma linguagem de programação simples.</p>		

## Provas escritas.

### Atividades avaliativas:

#### Primeira avaliação (1/3 da nota total)

- Temas: introdução, análise léxica, análise sintática descendente
- Prova escrita 1 (21/09)

#### Segunda avaliação (1/3 da nota total)

- Temas: análise sintática ascendente, análise semântica
- Prova escrita 2 (15/10)

#### Terceira avaliação (1/3 da nota total)

- Tema: implementação do *front end* de um compilador
- Exercício prático 1: analisador léxico (entregar até 10/09)
- Exercício prático 2: analisador sintático (entregar até 01/10)
- Exercício prático 3: analisador semântico (entregar até 15/10)

#### Exame especial

- Prova escrita incluindo conteúdo teórico e/ou prático
- Exame parcial substitui uma das três avaliações
- Exame total substitui todas as três avaliações

### Cronograma:

Aula	Data	Dia	Síncrona	Conteúdo
1	24 ago	seg	sim	Apresentação da disciplina; Compilação e interpretação
2	25 ago	ter		Gramáticas. Linguagem fonte
3	26 ago	qua		Linguagem de implementação; Ferramentas
4	27 ago	qui		Análise léxica: introdução; analisador <i>ad hoc</i>
5	31 ago	seg		Análise léxica: geradores
6	01 set	ter		Análise léxica: projeto
7	02 set	qua		Análise léxica: expressões regulares
8	03 set	qui		Análise léxica: autômatos finitos
	07 set	seg		<i>Feriado</i>
9	08 set	ter		Análise léxica: analisador baseado em autômatos finitos
10	09 set	qua		Análise sintática: introdução
11	10 set	qui		Análise sintática: gerador
12	14 set	seg		Análise sintática: projeto
13	15 set	ter		Análise sintática: análise descendente recursiva
14	16 set	qua		Análise sintática: <i>nullable</i> , conjuntos <i>first</i> e <i>follow</i>
15	17 set	qui		Análise sintática: tabelas LL(0) e LL(1)
16	21 set	seg	sim	<b>Prova 1</b>
17	22 set	ter		Análise sintática: análise ascendente e autômato de pilha
18	23 set	qua		Análise sintática: tabelas LR(0) e SLR
19	24 set	qui		Análise sintática: tabela LR(1)
20	28 set	seg		Análise sintática: tabela LALR(1)

21	29 set	ter		Análise sintática: resolução de conflitos
22	30 set	qua		Análise sintática: hierarquia de gramáticas
23	01 out	qui		Árvores de sintaxe abstrata
24	05 out	seg		Análise semântica: introdução
25	06 out	ter		Análise semântica: projeto
26	07 out	qua		<i>Feriado</i>
27	08 out	qui		Análise semântica: tabela de símbolos
	12 out	seg		Análise semântica: tipagem
28	13 out	ter		Análise semântica: regras de análise semântica
29	14 out	qua		Análise semântica: escopo
30	15 out	qui	sim	<b>Prova 2</b>
	21 out	qua	sim	<b>Exame especial</b>

#### Bibliografia Básica:

- A. V. Aho, M. S. Lam, R. Sethi, J. D. Ulman. **Compiladores: Princípios, Técnicas e Ferramentas**. 2. edição. Pearson, 2007. [Disponível online em MinhaUFOP/BVirtual Pearson](#).
- Santos, Pedro Reis; Langlois, ThiBault. **Compiladores: da Teoria à Prática**. LTC, 2018. [Disponível online em MinhaUFOP/Minha Biblioteca..](#)
- Louden C. **Compiladores: Princípios e Prática**. Cengage, 2004. [Disponível online em MinhaUFOP/Minha Biblioteca..](#)

#### Bibliografia Complementar:

- *Observação*: Só há três títulos de compiladores disponíveis na Biblioteca Digital do sistema MinhaUFOP.
- Appel, Andrew W. **Modern Compiler Implementation in ML**. Cambridge University Press, 1998.
- Mogensen, Torben Ægidius. **Introduction to Compiler Design**. 2. edição. Springer, 2011.
- Grüne, Dick; Reeuwijk, Kees van; Bal, Henri E.; Jacobs, Criel J. H.; Langendoen, Koen. **Modern Compiler Design**. 2. edição. Springer, 2012.
- Wilhelm, Reinhard; Seidl, Helmut; Hack, Sebastian. **Compiler Design: Syntactic and Semantic Analysis**. Springer, 2013.
- Fischer, Charles N.; Cytron, Ron K.; LeBlanc, Richard J. Jr. **Crafting a Compiler**. Pearson, 2010.
- Cooper, Keith D; Torczon, Linda. **Engineering a Compiler**. 2. edição. Elsevier, 2011. ISBN-13: