

## Plano de Ensino e Cronograma

### 1. Objetivos

Apresentar conceitos fundamentais sobre o processo de compilação, implementação de linguagens, técnicas de projeto e construção de compiladores. Capacitar o aluno a identificar aplicações potenciais em problemas não diretamente relacionados ao projeto de compiladores para linguagens de propósito geral.

### 2. Ementa

Processos de compilação e interpretação. Análise Léxica. Análise Sintática. Tabelas de símbolos. Análise semântica. Tratamento e recuperação de erros. Tradução dirigida por sintaxe. Geração de Código. Técnicas fundamentais de otimização de código. Ambientes em tempo de execução. Ambientes de interpretação.

### 3. Programa

#### I. Introdução

1. Conceitos [A1.1, A1.4, B1.1, B1.2, B3.1, B3.3, C1.1, C1.2, C3.1, C3.3]
2. O processo de tradução [A1.3, A1.5, B1.3, C1.3]
3. Programas relacionados aos compiladores [A1.2, B1.4, C1.4]

#### II. Análise Léxica

1. Objetivos da análise léxica [B3.1, A2.1, C3.1]
2. Padrão de formação de lexemas [B3.3, A2.2, C3.3]
3. Tabelas de símbolos [B2.7, A6.3, C7.6]
4. Implementação [B3.4, A2.3.3, C3.4]

#### III. Análise Sintática

1. Objetivos da análise sintática [A3.1, A4.1, B4.1, C4.1]
2. Gramáticas livres de contexto preditivas [A3.2-5, A4.2, B4.2-4, D, C4.2-4]
3. Parsing LL e LR [B4.4-4.7.3, D4, A3.1, C4.4]
4. Recuperação de Erros [B4.1, A4.5, C4.1]
5. Implementação [B4.4, A4.1, C4.4]

#### IV. Tradução Dirigida por Sintaxe

1. Atributos herdados e sintetizados [A6.1, A6.2.2, A6.2.3, B5.1, C5.1]
2. Esquemas de tradução [B5.4, A6.1, C5.4]
3. Implementação [B5.5, A6.2.2, C5.5]

#### V. Análise Semântica

1. Objetivos da análise semântica [B6, A6, C6]
2. Verificação de unicidade [B6, A6.3.3, C6]
3. Verificação de classes [B6, A6.3.2, C6]
4. Verificação de tipos [B6.3, B6.5, A6.4.1, A6.4.2, C6.1-2]

#### VI. Geração de Código

1. Objetivos da geração de código [B6.1, B8.1, A8, C9.1]
2. O conjunto de instruções [B8.1, A8, C9.2]
3. Declarações [B6.3, J, A8, C8.2]
4. Expressões [B6.4, J, A8, C8.3-4]
5. Comandos [B8.4, A8.4, A8, C8.4]
6. Introdução à otimização de código [B8.7]

#### VII. Ambientes em Tempo de Execução

1. Organização de memória [B7.2, A7.1, A7.4.3, C7.2]
2. Estratégias de alocação de memória [B7.3, A7.2, C7.3]
3. Pilha de ativação [B7.2, A7.2, C7.1]
4. Acesso a nomes não-locais [B7.3, A7.3, C7.4]
5. Passagem de parâmetros [B7.5, A7.5.1, A7.5.2, C7.5]

### 4. Avaliação da Aprendizagem

- |                         |           |
|-------------------------|-----------|
| (a) Trabalhos práticos: | 30 pontos |
| (b) 2 testes:           | 60 pontos |
| (c) Artigo:             | 5 pontos  |
| (d) ADA:                | 5 pontos  |

### 5. Bibliografia

- [A] (livro-texto) Louden, Kenneth C. *Compiladores: Princípios e práticas*. Pioneira Thompson Learning, 2004.
- [B] (livro-texto) Aho, A., Sethi, R., Ullman, J. *Compiladores: Princípios, Técnicas e Ferramentas*. Pearson, 2008.
- [C] (livro-texto) Aho, A., Sethi, R., Ullman, J. *Compilers: Principles, Techniques and Tools*. Addison-Wesley, 1986. Tradução: *Compiladores: Princípios, Técnicas e Ferramentas*. Guanabara Koogan, 1995.
- [D] Sudkamp, Thomas A. *Languages and Machines: An introduction to the theory of computer science*. Addison Wesley, Reading, 1997.
- [E] Grune, D et al. Projeto moderno de compiladores: implementação e aplicações. Rio de Janeiro: Campus, 2001.
- [F] Aho, A., Ullman, J. *The Theory of Parsing, Translation and Compiling*. Prentice-Hall, 1973.
- [G] T. Kowaltowski. *Implementação de Linguagens de Programação*. Guanabara Dois, 1983.

[H] C. N. Fischer, R. J. LeBlanc. *Crafting a Compiler*. The Benjamin/Cummings Publishing Company, 1988.  
 [I] Ullman, J. *Fundamental Concepts of Programming Systems*. Addison-Wesley, 1976.  
 [J] Hyde, Randall. *The Art of Assembly Programming*. [http://webster.cs.ucr.edu/AoA/DOS/pdf/0\\_AoAPDF.html](http://webster.cs.ucr.edu/AoA/DOS/pdf/0_AoAPDF.html)

## 6. Observações

- Os alunos que perderem algum teste por motivo justificado, dentro das normas da Universidade, farão a reposição ao final do semestre com o mesmo conteúdo do exame perdido.
- O exame de reavaliação vale 100 pontos, dos quais 30 são aproveitados do trabalho. A nota final é a média entre a nota do exame e a do semestre.
- Não é possível o abono de faltas fora o previsto no “Manual do Aluno”. O aluno deve conferir o lançamento de falta até o final da aula. Atenção: O aluno que não comparecer a uma avaliação, seja ela escrita ou apresentação de trabalho, recebe falta.
- As avaliações devem ser feitas à caneta, letra legível (preferencialmente de forma) e tamanho semelhante ao do enunciado. Não serão corrigidas respostas fora dessas especificações. Os alunos não podem deixar a sala antes nem entrar após 20 minutos decorridos do início da prova.
- Pedidos de revisão de nota devem seguir o regulamento da Universidade.
- Trabalhos copiados de qualquer fonte ou iguais em parte ou totalidade não serão avaliados, independente de quem lesou ou foi lesado. É responsabilidade do grupo manter o sigilo sobre seu trabalho. O trabalho deve ser realizado exclusivamente pelos componentes do grupo, sem auxílio de terceiros.
- Será pedida ao Colegiado uma advertência formal para alunos que agirem com improbidade nas avaliações.
- Não é permitido gravar a aula nem fotografar ou reproduzir provas. O material disponibilizado no SGA é para uso próprio e não pode ser divulgado para terceiros ou usado para outros fins.
- Não é permitido o uso de celular durante as aulas.
- Durante as avaliações não é permitido o porte, uso ou acesso a celulares e outros dispositivos eletrônicos. Calculadoras quando permitidas não podem conter recursos de memória e/ou transmissão de dados. O uso desses tipos de aparelhos resultará na anulação da nota correspondente.

## 7. Cronograma

Aula	Data	Assunto
1	04/02	Apresentação do curso
2	06/02	Introdução – Conceitos básicos
3	11/02	Introdução – Conceitos básicos
4	13/02	O processo de compilação
5	18/02	Análise Léxica
6	20/02	Análise Léxica
7	25/02	Especificação do trabalho
8	27/02	Parsing descendente - Gramáticas LL(1) equivalentes
9	11/03	Parsing descendente - Gramáticas LL(1) equivalentes
10	13/03	Gramáticas LL(1) equivalentes
11	18/03	Exercícios
12	20/03	GREs, Precedência, Parser descendente e ascendente
13	25/03	Exercícios
14	27/03	<b>Teste 1</b>
15	01/04	Reserva
16	03/04	Tradução
17	08/04	Tradução
18	10/04	Implementação do tradutor
19	22/04	<b>Apresentação Trabalho</b>
20	24/04	<b>Apresentação Trabalho</b>
21	29/04	Análise Semântica
22	06/04	Análise Semântica
23	08/05	Geração de Código
24	13/05	Geração de Código
25	15/05	Geração de Código
26	20/05	Otimização de Código
27	22/05	Tempo de Execução
28	27/05	Tempo de Execução
29	29/05	Exercícios
30	03/06	Exercícios
31	05/06	<b>Teste 2</b>
32	10/06	<b>Apresentação trabalho</b>
33	12/06	<b>Teste 2ª. Chamada</b>
34	17/06	<b>Exame de Reavaliação</b>
35	19/06	<b>Reserva</b>

## 8. Artigo

O artigo a ser lido neste semestre é:

Xin Chen, David Pager. “LR(1) Parser Generator Hyacc”. In: *Proceedings of International Conference on Software Engineering Research and Practice*, p.471-477. Las Vegas, 2011.



**Pontifícia Universidade Católica de Minas Gerais**  
**ICEI - Departamento de Ciência da Computação**  
**Disciplina: Compiladores**  
**Prof. Alexei Machado**

O artigo deve ser acessado através do portal de periódicos da CAPES e será avaliado junto com o teste 1, em uma questão valendo 5 pontos.

**Contato: [alexeimcmachado@gmail.com](mailto:alexeimcmachado@gmail.com) (não envie mensagens pelo SGA)**