

**MINISTÉRIO DA EDUCAÇÃO**  
**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA**  
**DIRETORIA DE ENSINO (DIREN)**  
**DEPARTAMENTO DE ENSINO SUPERIOR (DEPES)**  
**DEPARTAMENTO DE INFORMÁTICA (DEPIN)**  
**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO (GCC)**

|                                                      |                              |
|------------------------------------------------------|------------------------------|
| DEPARTAMENTO                                         | PLANO DE CURSO DA DISCIPLINA |
| <b>DEPIN - Departamento Acadêmico de Informática</b> | <b>COMPILADORES</b>          |

|                |              |         |                            |                                 |
|----------------|--------------|---------|----------------------------|---------------------------------|
| CÓDIGO         | PERÍODO      | ANO     | SEMESTRE                   | PRÉ-REQUISITOS                  |
| <b>GCC1730</b> | 7º           | 2012    | 2                          |                                 |
| CRÉDITOS       | AULAS/SEMANA |         |                            | GCC 1207<br>Estruturas de Dados |
|                | TEÓRICA      | PRÁTICA | ESTÁGIO                    |                                 |
| 4              | 4            | 0       | 0                          |                                 |
|                |              |         | TOTAL DE AULAS NO SEMESTRE |                                 |
|                |              |         | 72                         |                                 |

| EMENTA                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Compiladores. Análise Léxica. Análise Sintática. Tradução Dirigida por sintaxe. Verificação de tipos. Geração de Código Intermediário. Otimização de Código. Geração de Código Objeto. Ferramentas para implementação de Compiladores. |

| BIBLIOGRAFIA                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Bibliografia básica</p> <ol style="list-style-type: none"> <li>1. AHO, A. V., LAM, M. S., SETHI, R. e ULLMAN, J. D. <i>Compiladores: princípios, técnicas e ferramentas</i>. Ed. Pearson.</li> <li>2. PRICE, A. M. A. e TOSCANI, S. S. <i>Implementação de Linguagens de Programação: Compiladores</i>. Série Livros Didáticos UFRGS. Ed. Bookman.</li> <li>3. LOUDEN, K. C. <i>Compiladores: princípios e práticas</i>. Ed. Thomson.</li> </ol> <p>Bibliografia complementar</p> <ol style="list-style-type: none"> <li>1. MENEZES, P. B. <i>Linguagens Formais e Autômatos</i>. Série Livros Didáticos UFRGS. Ed. Bookman.</li> <li>2. GRUNE, D., BAL, H. E., JACOBS, C. J. H. e LANGENDOEN, K. G. <i>Projeto Moderno de Compiladores: implementação e aplicações</i>. Ed. Campus.</li> <li>3. COOPER, K. D. e TORCZON, L. <i>Construindo Compiladores</i>. Ed. Campus/Elsevier.</li> <li>4. SEBESTA, R. W. <i>Conceitos de Linguagens de Programação</i>. Ed. Bookman.</li> <li>5. CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L. e STEIN, C. <i>Algoritmos – teoria e prática</i>. Ed. Campus/Elsevier</li> </ol> |

| OBJETIVO GERAL                                                                                                                                          |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fornecer conhecimento sobre a teoria e prática para construção de compiladores, bem como conhecimento das técnicas de compilação em problemas diversos. |

## METODOLOGIA

- Aulas expositivas com recursos audiovisuais.

## CRITÉRIO DE AVALIAÇÃO

A avaliação semestral envolve duas provas escritas (P1 e P2). As datas das provas são agendadas entre o professor e a turma. A média parcial (MP) será calculada pelo cômputo da média aritmética simples entre a nota P1 e P2:

$$MP = (P1 + P2) / 2$$

O aluno que faltar a uma das duas provas terá direito a uma avaliação alternativa, denominada segunda chamada, versando sobre todos os tópicos abordados no curso, e cuja data também é agendada entre docente e discentes. A nota obtida nessa 2ª chamada substituirá a da avaliação P1 ou P2 onde o aluno não esteve presente. Caso ele falte às duas avaliações, terá atribuído o grau ZERO em uma delas.

Segundo o regimento do CEFET-RJ, caso o aluno obtenha média parcial inferior a 3,0 (três e zero) estará reprovado diretamente. Graus MP maiores ou iguais a 7,0 (sete e zero) aprovam diretamente o aluno. Em situações onde o aluno tenha grau MP entre 3,0 inclusive e 7,0 exclusive, terá direito a uma prova final (PF), que, juntamente com a média parcial gerará uma nova média, denominada média final (MF). Essa média é calculada da seguinte forma:

$$MF = (MP + PF) / 2$$

Para ser aprovado, o aluno deve alcançar uma MF maior ou igual a 5,0 (cinco e zero). Caso contrário, estará reprovado, devendo repetir a componente curricular.

## CHEFE DO DEPARTAMENTO

| NOME | ASSINATURA |
|------|------------|
|      |            |

## PROFESSOR RESPONSÁVEL PELA DISCIPLINA

| NOME | ASSINATURA |
|------|------------|
|      |            |

## PROGRAMA

1. Visão Geral da Compilação
  - 1.1. Tradutores de linguagem de programação
  - 1.2. Visão geral do tradutor
2. Análise Léxica (*Scanner*)
  - 2.1. Objetivo do analisador léxico
  - 2.2. Reconhecimento de palavras (*tokens*)
  - 2.3. Construção da tabela de símbolos
  - 2.4. Da expressão regular ao *Scanner*
  - 2.5. Implementação
  - 2.6. Principais erros de compilação na análise léxica

### 3. Análise Sintática (*Parser*)

#### 3.1. Objetivo do analisador sintático

#### 3.2. Análise sintática descendente (*top-down*)

##### 3.2.1. Árvore de derivação (gramatical)

##### 3.2.2. Derivação mais à esquerda e mais à direita

##### 3.2.3. Gramática ambígua

##### 3.2.4. Transformações da Gramática Livre do Contexto

###### 3.2.4.1. Eliminação de recursividade à esquerda

###### 3.2.4.2. Eliminação do vazio

###### 3.2.4.3. Fatoração da gramática

##### 3.2.5. Análise recursiva com retrocesso

##### 3.2.6. Análise recursiva preditiva

##### 3.2.7. Análise preditiva tabular – LL(1)

#### 3.3. Análise sintática redutiva (*bottom-up*)

##### 3.3.1. Analisadores de precedência de operadores

##### 3.3.2. Funções de precedência

##### 3.3.3. Analisador LR(1)

#### 3.4. Tratamento e recuperação de erros

### 4. Análise Semântica

#### 4.1. Verificação de tipos

#### 4.2. Tradução dirigida por sintaxe

#### 4.3. Esquemas de tradução

#### 4.4. Caminhamento na árvore de derivação

#### 4.5. Gramática de atributos

#### 4.6. Árvore de sintaxe

### 5. Geração de Código Intermediário

#### 5.1. Notação pós-fixa e prefixa

#### 5.2. Código de três endereços

#### 5.3. Esquema de código para comandos de atribuição

#### 5.4. Esquema de código para expressões aritméticas

#### 5.5. Esquema de código para expressões lógicas

#### 5.6. Esquema de código para comandos *while* e *if*

### 6. Otimização de Código

#### 6.1. Representação de bloco básico por meio de grafos

#### 6.2. Principais algoritmos de otimização de código intermediário

### 7. Geração de Código Objeto

#### 7.1. Alocação de registradores

#### 7.2. Máquina objeto

### 8. Geradores de analisadores léxicos e sintáticos