

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E DE ESTATÍSTICA

PLANO DE ENSINO

I. Identificação:

1. **Disciplina:** INE5622 - Introdução a Compiladores
2. **Turma(s):** 06238
3. **Carga horária:** 72 horas-aula Teóricas: 54 Práticas: 18
4. **Período:** 2º semestre de 2013
5. **Curso(s):** Sistemas de Informação (238)
6. **Requisito(s):** INE5609 - Estruturas de Dados

II. Ementa: Gramáticas, Autômatos e Linguagens; Análise Léxica; Análise Sintática; Análise Semântica e Geração de Código.

III. Objetivos:

III.1 - Geral: Conhecer o processo de especificação e implementação de linguagens de programação, a partir dos conceitos, modelos, técnicas e ferramentas que compõem a Teoria das Linguagens Formais e a Teoria de Compiladores.

III.2 - Específicos:

1. Adquirir uma visão geral sobre o Processo de Compilação sob o ponto de vista de implementação.
2. Adquirir noções básicas sobre a Teoria das Linguagens Formais.
3. Saber especificar aspectos léxicos e sintáticos de linguagens através de autômatos e gramáticas.
4. Conhecer critérios e características usados no projeto/avaliação de Linguagens de Programação.
5. Conhecer as principais técnicas e ferramentas de apoio usadas na construção de compiladores, sabendo usá-las na especificação e implementação de linguagens de programação.
6. Obter Subsídios que permitam um melhor entendimento, utilização e avaliação das Linguagens de Programação.

IV. Conteúdo Programático, Metodologia e Carga Horária

Conteúdo Programático	Metodologia	Carga Horária
1. Introdução a Compiladores e Introdução a Teoria das Ling. Formais	AEX	6
2. Gramáticas: Motivação. Definição formal. Derivação e redução. Linguagens. Tipos de gramáticas e linguagens. Sentença vazia.	AEX	10
3. Linguagens de Programação: Conceitos fundamentais; Paradigmas; Critérios para avaliação e projeto. Especificação. Ferramentas de implementação.	OTR	4
4. Análise Léxica: Autômatos Finitos Determinísticos e Não-Determinísticos, Determinização e Minimização, Relação entre AF e GR. Conjuntos Regulares e Expressões Regulares. Aspectos Léxicos de Ling. de Programação. Especificação/implementação de um Analisador Léxico.	AEX/APR	10
5. Análise Sintática – Gramáticas Livre de Contexto (GLC) : Árvore de derivação, ambigüidade, transformações e simplificações. Técnicas de AS Ascendentes e Descendentes. Aspectos Sintáticos de Ling. de Prog.. Especificação/Implementação de um Analisador Sintático.	AEX/APR	12
6. Análise Semântica – Aspectos semânticos de Linguagens de Programação. Modelos	AEX/APR	16

de Especificação. Regras e Ações Semânticas. Especificação e implementação de um Analisador Semântico.		
7. Geração de código Intermediário, Otimização e Interpretação.	AEX/APR	14

(AEX=Aula Expositiva; LAB=Aula de laboratório; APR=Aula prática; OTR=Outros).

V. SISTEMA DE AVALIAÇÃO

V.1-Critério para Aprovação: Será aprovado na disciplina o aluno que obtiver Média Final (MF) igual ou superior a 6.0 e frequência igual ou superior a 75%. A MF será calculada através da fórmula:

$$MF = (P1 + P2 + P3)/3 * 5.0 + IC1 + IC2*2.5 + MTE*1.5$$

Onde: P1 = Prova1, P2 = Prova2, P3 = Prova3, IC1 = Imp. do compilador - parte 1, IC2 = Imp. do compilador - parte 2, MTE = (TT + MLE)/2 onde TT = Trabalho Teórico e MLE = Média das Listas de Exercícios

V.2-Recuperação: Se MF for maior ou igual 3.0 e menor que 6.0, o aluno poderá fazer uma prova de recuperação (REC) e a Nota Final (NF) será $NF = (MF + REC)/2$. Caso contrário, $NF = MF$.

V.3 - Cronograma de Avaliação:

Tópico Avaliado	Forma	Semana Provável
Itens 1 a 3	Prova 1	7 ^a
Itens 4 e 5	Prova 2	11 ^a
Itens 6 e 7	Prova 3	17 ^a
Itens 1 a 7	Listas de Exercícios	variável
Item 3	Trabalho Teórico	8 ^a
Itens 4 e 5	Implementação do Compilador (parte1)	13 ^a
Itens 6 e 7	Implementação do Compilador (parte2)	17 ^a
Itens 1 a 7	Recuperação	18 ^a

VI - Bibliografia Básica

1. Notas de aula
2. FURTADO, O. J. V. Apostila de Linguagens Formais e Compiladores, versão 2 - UFSC, 2002, disponível em www.inf.ufsc.br/~olinto
3. AHO, A.V.; SETHI, R. ULLMAN, J.D. Compiladores – Princípios, Técnicas e Ferramentas, Ed. Addison Wesley 2008 / LTC, 1995.

VII – Bibliografia Complementar

1. FISCHER, C. N., CYTRON, R. K., LEBLANC Jr., R. J. Crafting a Compiler, Addison Wesley, 2010.
2. SEBESTA, R.W., Conceitos de Linguagens de Programação, ed. Bookman, 5. edição, 2003.
3. RAMOS, M. V. M., NETO, J. J., VEGA, I. S., Linguagens Formais: Teoria, Modelagem e Implementação. Ed. Bookman, 2009.
4. HOPCROFT, J. F., ULLMAN, J. D., MOTWANI, R. Introdução à Teoria de Autômatos, Linguagens e Computação, Tradução da segunda edição Americana, Elsevier Editora Ltda, 2003.
5. PRICE, A. M. A., TOSCANI, S. S., Implementação de Ling. de Programação: Compiladores. Ed Sagra Luzzatto, 2008.
6. WILHELM, Reinhard; MAURER, Dieter. Compiler Design. Ed. Addison-Wesley, 1995, 606p.
7. AHO, A.V.; ULLMAN, J.D. Principles of Compiler Design. Ed Addison-Wesley, 1979, 603p.
8. LOUDEN, K. C.. Compiladores - Princípios e Práticas. Editora Thomson, 2004.
9. WATT, D., Programming Language Design Concepts, John Wiley and Sons, Ltd, 2004.
10. APPLEBY, D., Programming Languages – Paradigm and Practice. Ed McGraw-Hill, Inc., 1991.