



PLANO DE ENSINO

1. IDENTIFICAÇÃO

Curso: Ciência da Computação - Noturno

Componente Curricular: Construção de Compiladores (turma 11710)

Fase: 7 (sete)

Ano/Semestre: 2015/2

Número da turma: 11710

Numero de Créditos: 4 (quatro)

Carga horária - Hora Aula: 72 horas

Carga horária - Hora Relógio: 60 horas

Professor: Braulio Adriano de Mello

Atendimento aos estudantes: Sabados entre 08:30 e 10:10 (com confirmação de atendimento) e nos períodos que antecedem os horários de aula conforme disponibilidade dos estudantes.

2. OBJETIVO GERAL DO CURSO

O curso tem por objetivo a formação integral de novos cientistas e profissionais da computação, os quais deverão possuir conhecimentos técnicos e científicos e serem capazes de aplicar estes conhecimentos, de forma inovadora e transformadora, nas diferentes áreas de conhecimento da Computação. Adicionalmente, os egressos do curso deverão ser capazes de adaptar-se às constantes mudanças tecnológicas e sociais, e ter uma formação ao mesmo tempo cidadã, interdisciplinar e profissional.

3. EMENTA

A disciplina de Construção de Compiladores permite ao estudante compreender conceitos e características do processo de compilação. Oferece melhores condições para avaliar as propriedades de uma linguagem e seus efeitos nos códigos gerados considerando, por exemplo, a relação com chamadas ao sistema (sistemas operativos), desdobramentos dos conceitos de linguagens de programação e computabilidade. Podem ser também enumeradas benefícios indiretos resultantes de conhecimentos em torno do processo de compilação. Entre eles, a disponibilidade de referenciais ou informações adicionais que interferem positivamente nos métodos ou práticas de programação.

4. OBJETIVOS

4.1. GERAL

Compreender a estrutura de compiladores, o processo de compilação no reconhecimento de linguagens e a geração/otimização de código, construção de analisadores léxicos e sintáticos.

4.2. ESPECÍFICOS

- Estudo da Estrutura e Processo de compilação
- Entender e construir analisadores observando formalismos e algoritmos utilizados

para os diferentes tipos de máquinas reconecedoras

- Estudo e experimentação de soluções para verificação semântica e otimização de código
- Desenvolver um projeto prático atendendo etapas do processo de compilação incentivando os estudantes a realizar esforços integradores com demais áreas do curso, preferencialmente observando os CCRs em curso no semestre;
- Fortalecer iniciativas tais como elaboração de novos materiais de apoio e dinâmicas alternativas de atividades buscando melhorar as condições didático-pedagógicas para aprendizado de compiladores.

5. CRONOGRAMA E CONTEÚDOS PROGRAMÁTICOS

ENCONTRO	CONTEÚDO
1	Apresentação da disciplina, apreciação do plano de ensino e sistema de avaliação
2	Linguagens Formais e Autômatos
3	Componentes de um Compilador e processo de compilação: Analisadores Léxico, Sintático e Semântico e Gerador de Código. Recuperação de estudos e Avaliação PEA
4	Interpretação/compilação, analisadores, geração e otimização de código
5	Construção de analisadores léxicos: autômato finito como máquina reconhedora. TF1 .
6	Estudo de caso: Lex e yacc. Recuperação de estudos e Avaliação PEA
7	Projeto 1: Especificação e implementação de reconhedor léxico.
8, 9, 10, 11	Processo de construção de analisadores léxicos. Desenvolvimento e apresentação (no encontro 11) do Projeto 1.
12, 13	Análise sintática: Analisadores ascendentes e descendentes, simplificação de GLCs, árvores de derivação (esq., dir.), analisadores LR e LL.
14,15	Autômatos de Pilha como máquina reconhedora. Recuperação de estudos e Avaliação PEA
16	Análise sintática: precedência de operadores (shift-reduce).
17, 18	Implementação de analisador sintático para a gramática de operadores com shift-reduce. TF2 .
19	Análise sintática SLR: construção de itens válidos e conjunto de transições (gramática de operadores). Recuperação de estudos e Avaliação PEA
20	Análise sintática SLR: construção da tabela de parsing.
21	Algoritmo de reconhecimento sintático SLR. Recuperação de estudos e Avaliação PEA .
22	Projeto 2: Implementação de reconhedor sintático SLR.
23, 24, 25, 26, 27, 28	Processo de construção de analisadores sintáticos. Desenvolvimento e apresentação (no encontro 28) do Projeto 2.
29,30	Análise Semântica e Código intermediário: conceitos, características, tradução dirigida por sintaxe.
31,32	Otimização de código: conceitos, características, otimização por Grafos Acíclicos Dirigidos. TF3 .
33,34	Código objeto e editor de ligação. Recuperação de estudos e Avaliação PEA .
35	Recuperação de estudos e Avaliação complementar.
36	Apresentação dos TFs.

Obs.: As atividades/cronograma dos encontros podem ser ajustados no decorrer do semestre.

6. PROCEDIMENTOS METODOLÓGICOS

Embora a disciplina contemple conteúdo a ser trabalhado em encontros teóricos, maior esforço é dedicado ao estudo do processo de construção de compiladores. Deste modo, serão contempladas aulas expositivas/dialogadas para atendimento de conceitos específicos e atividades práticas no projeto e implementação dos processos conforme a estrutura de compiladores. O perfil aplicado da disciplina aponta para maior ênfase em exercícios práticos, demonstrações, contextualização experimentação e implementação.

7. AVALIAÇÃO DO PROCESSO ENSINO-APRENDIZAGEM

A avaliação do processo ensino-aprendizagem conta com, pelo menos, 8 oportunidades distribuídas ao longo do período. Seis avaliações (AvaliaçõesPEA) que podem fazer uso de métodos diferenciados, como por exemplo: provas teóricas, exercício prático, seminários curtos, elaboração de textos técnicos, entre outros. Cada AvaliaçãoPEA utiliza em torno de 25% do tempo do encontro. Dois projetos de média duração e complexidade que incluem parte prática (implementação) e escrita (artigo técnico). Os Projetos representam 60% e as AvaliaçõesPEA representam 40% da nota final

Avaliação de projetos: 50% da nota pela parte escrita e implementação (grupo) e 50% da nota de acordo com avaliação individual (clareza, domínio, conhecimento de detalhes e etapas do desenvolvimento) independente do grupo de trabalho.

NP1: Média das 3 primeiras AvaliaçõesPEA (peso 0,20)

NP2: Projeto 1 (peso 0,30)

NP3: Média das 3 últimas AvaliaçõesPEA (peso 0,20)

NP4: Projeto 2 (peso 0,30)

Avaliação complementar prevista para o final do semestre pode substituir a menor nota parcial (NP).

Os TFs facultam o acréscimo de até 5 pontos (num total de 100) na média final após correção e se necessários e suficientes para alcançar a nota mínima para aprovação.

7.1 RECUPERAÇÃO: NOVAS OPORTUNIDADES DE APRENDIZAGEM E AVALIAÇÃO

Ao final do semestre, caso não tenha sido alcançada pontuação suficiente para aprovação, o estudante poderá usufruir de duas oportunidades de recuperação:

a) Avaliação complementar (todo o conteúdo do semestre) cuja nota será utilizada para substituir a menor nota dentre as NPs.

b) Trabalhos facultativos (TF) individuais que, se entregues até o último dia letivo de aula da disciplina, podem ser utilizados para aumento da média final limitado a 5 pontos num total de 100. Trabalhos não realizados não implicam em redução de nota.

Todas as AvaliaçõesPEA são precedidas de oportunidades para recuperação de estudos e se apresentam, também, como oportunidade de recuperação de rendimento ao dividir o conteúdo em múltiplos momentos de avaliação.

O horário extra-classe para atendimento de estudantes também oportuniza a recuperação de estudos.

8. REFERÊNCIAS

8.1 BÁSICA

PRICE, A. M. A., TOSCANI, S. S. Implementação de Linguagens de Programação: Compiladores. Bookman Companhia Ed., 2008.

HOPCROFT, J. F., ULLMAN, J. D., Motwani, R., "Introdução a teoria dos automatos", Ed. Campus, 2002.

AHO, A. V., SETHI, R., LAM, M., "Compiladores: Princípios, técnicas e ferramentas", Ed. Longman do Brasil, 2007.

GRUNE, D., BAL, H. E., JACOBS, C., LANGENDOEN, K. Projeto Moderno De Compiladores: implementação e aplicações. Rio de Janeiro: Campus, 2001.

8.2 COMPLEMENTAR

WOOD, D. , “**Theory of Computation**”, Ed. John Wiley & Sons, 1987.

FURTADO, O. J. V., “**Apostila de Linguagens Formais e Compiladores – versão 2**”, UFSC, 2002.

DELAMARO, M. E. **Como Construir Um Compilador Utilizando Ferramentas Java**. Rio de Janeiro: NOVATEC, 2004.

LOUDEN, K. C. **Compiladores Princípios e Práticas**. São Paulo: THOMSON PIONEIRA, 2004.

Furtado, Olinto. Linguagens formais e compiladores. <www.inf.ufsc.br/~olinto/apostila-lfc.doc>. Acesso em 30/09/2012.

8.3 SUGESTÕES

Professor

Coordenador do curso