

**UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO  
PROGRAMA DE DISCIPLINA**

<b>Nome</b>		<b>Colegiado</b>	<b>Código</b>	<b>Semestre</b>
Compiladores		CECOMP	CCMP0051	2016.2
<b>Carga Horária</b>	Teórica: 60	Prática: 0	Horários: Seg 16:00 às 18:00 e Qua 16:00 às 18:00	
<b>Curso Atendido</b>				<b>Turma</b>
Engenharia da Computação				C7
<b>Professor Responsável</b>				<b>Titulação</b>
Marcus Vinícius Midena Ramos				DS
<b>Ementa</b>				
Linguagens-fonte, objeto, de alto-nível e de baixo-nível. Especificação de linguagens de programação. Compilação e interpretação. Processadores de linguagens de programação. Máquinas reais e virtuais. Bootstrapping. Análise sintática. Análise de contexto. Ambientes de execução. Geração de código. Otimização de código independente de máquina. Otimização de código dependente de máquina.				
<b>Objetivos</b>				
Capacitar o aluno na síntese, análise e manipulação de especificações de linguagens de programação de alto-nível, assim como no emprego de técnicas de implementação de processadores de linguagens. Capacitar o aluno na análise, síntese e manipulação de especificações de linguagens de programação; Fazer com que o aluno seja capaz de compreender os mecanismos internos e os princípios de funcionamento dos principais tipos de processadores de linguagens; Tornar o aluno capaz de avaliar, projetar e implementar processadores para linguagens de alto-nível; Permitir que o aluno compreenda o funcionamento de um sistema de execução típico, incluindo as interfaces com o sistema operacional e a arquitetura de hardware.				
<b>Metodologia</b>				
Aulas expositivas com realização de exercícios em sala de aula após a apresentação de cada novo conteúdo. Exercícios e atividades extra-classe (construção de um compilador).				
<b>Formas de Avaliação</b>				
Duas provas e projeto. A nota final será igual à média aritmética destas notas (peso 3 para cada uma das provas e peso 4 para o projeto).				

<b>Conteúdos Didáticos</b>				
<b>Data</b>	<b>Temas abordados/atividades desenvolvidas</b>	<b>Professor</b>	<b>Carga Horária</b>	
			<b>Teórica</b>	<b>Prática</b>
20/09	Apresentação e motivação Linguagens de alto e baixo-nível Linguagens fonte e objeto Compilação e interpretação Tipos de processadores Métodos para especificação de linguagens	Marcus	2:00h	-
22/09	Processadores de linguagens Diagramas-T Representação de processos	Marcus	2:00h	-
27/09	Compilador portátil Bootstrapping Desenvolvimento incremental de linguagens	Marcus	2:00h	-
29/09	Especificação de linguagens Revisão de gramáticas livres de contexto Sintaxe livre de contexto da linguagem mini-Triangle	Marcus	2:00h	-
04/10	Sintaxe livre de contexto x sintaxe dependente de contexto Sintaxe dependente de contexto da linguagem mini-Triangle Semântica da linguagem mini-Triangle Fases de compilação Front-end e back-end	Marcus	2:00h	-
06/10	Exemplo em mini-Triangle: programa-fonte, AST, AST decorada e programa-objeto Passos de compilação Um passo e vários passos: vantagens e desvantagens	Marcus	2:00h	-

11/10	Estrutura léxica Símbolos úteis e inúteis Tokens, tipo e valor Alfabeto1 -> linguagem1 = alfabeto2 -> linguagem2 Expressões regulares, BNF e EBNF	Marcus	2:00h	-
13/10	Substituição de não-terminais Fatoração à esquerda Eliminação de recursão à esquerda Starter Reduções e derivações Análise sintática descendente e ascendente	Marcus	2:00h	-
18/10	Gramáticas e linguagens LL(k) e LR(k) Definição de gramática LL(1) Gramáticas LL(1) sem regras vazias Gramáticas LL(1) com regras vazias	Marcus	2:00h	-
25/10	Exemplos e exercícios Gramática léxica Gramática sintática	Marcus	2:00h	-
27/10	Método recursivo descendente Aplicação do método recursivo descendente à construção de analisadores sintáticos	Marcus	2:00h	-
01/11	Aplicação do método recursivo descendente à construção de analisadores léxicos Aspectos de implementação	Marcus	2:00h	-
03/11	Desenvolvimento do projeto e esclarecimento de dúvidas	Marcus	2:00h	-
08/11	Montagem da AST Análise de contexto: identificação Tabela de identificação: atributos e métodos Linguagens com estrutura de bloco monolítica	Marcus	2:00h	-
17/11	Prova 1	Marcus	2:00h	-
22/11	Linguagens com estrutura de blocos planos Linguagens com estrutura de blocos aninhados Escopo estático x escopo dinâmico	Marcus	2:00h	-
24/11	Verificação de tipos Tipo estático x tipo dinâmico Implementação de analisadores de contexto Padrão de projeto Visitor	Marcus	2:00h	-
29/11	Representação de dados Tipos simples Tipos agregados heterogêneos	Marcus	2:00h	-
01/12	Tipos agregados homogêneos Verificação de índices Origem virtual Múltiplas dimensões	Marcus	2:00h	-
06/12	Exemplos de representação de dados Avaliação de expressões em máquinas de registradores Avaliação de expressões em máquinas de pilha Instruções LOAD, LOADL, STORE e CALL	Marcus	2:00h	-
08/12	Alocação de memória - estática Alocação de memória - automática Estrutura do stack-frame (SB, LB, ST, L1, L2, ..., LD, LE, ER)	Marcus	2:00h	-
13/12	Passagem de parâmetros Retorno de valor de função Exemplo completo	Marcus	2:00h	-
15/12	Instruções CALL, RETURN, LOADA, LOADI, STOREI, PUSH e POP Passagem de parâmetros por endereço x por valor Exemplo completo	Marcus	2:00h	-

20/12	Cálculo dos registradores de acesso às variáveis e do link estático Funções de código Padrões de código Exemplos de padrões de código Instruções JUMP, JUMPIF	Marcus	2:00h	-
22/12	Exemplos de aplicação de funções e padrões de código Implementação dos métodos de geração de código	Marcus	2:00h	-
17/01	Padrão de código para declaração de procedimentos e funções, comandos for e switch; Organização da memória, pilha e heap.	Marcus	2:00h	-
19/01	Revisão Visitor Exemplo funcional completo com análise sintática, impressão da árvore de sintaxe, análise de contexto e geração de código (códigos fonte e objeto).	Marcus	2:00h	-
24/01	Orientação de projeto	Marcus	2:00h	-
26/01	Conclusões: Ciclo de vida de linguagens de programação Detecção e recuperação de erros Eficiência da execução Interpretores Geradores de compiladores	Marcus	2:00h	-
31/02	Prova 2	Marcus	2:00h	-
02/02	Segunda Chamada.	Marcus	2:00h	-
09/02	Prova Final.	Marcus	2:00h	-
<b>Referências Bibliográficas</b>				
<ul style="list-style-type: none"> <li>• WATT D.A.; BROWN, D.F. Programming Language Processors in Java - Compilers and Interpreters. Pearson Education, 2000.</li> <li>• AHO, A.V. et al. Compiladores - Princípios, Técnicas e Ferramentas. 2ª ed. Pearson Education, 2007.</li> <li>• PRICE, A.M.A.; TOSCANI, S.S. Implementação de Linguagens de Programação - Compiladores. Bookman, 2008.</li> <li>• RICARTE I. Introdução à Compilação. Elsevier, 2008.</li> <li>• DELAMARO, M.E. Como Construir um Compilador - Utilizando Ferramentas Java. Novatec, 2004.</li> <li>• LOUDEN, K.C. Compiladores - Princípios e Práticas. Cengage Learning, 2004.</li> <li>• GRÜNE, H.D. et al. Projeto Moderno de Compiladores - Implementação e Aplicações. Elsevier, 2001.</li> </ul>				
Data: 23/09/2016				