



UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO  
CONSELHO UNIVERSITÁRIO

		UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO PROGRAMA DE DISCIPLINA	
<b>NOME</b>			
Compiladores		<b>COLEGIADO</b>	<b>CÓDIGO</b>
		CECOMP	CCMP0051
<b>SEMESTRE</b>		2019-2	
<b>CARGA HORÁRIA</b>	<b>TEÓR: 60</b>	<b>PRÁT: 0</b>	<b>HORÁRIOS:</b> Terças e quintas das 10:00h às 12:00h.
<b>CURSOS ATENDIDOS</b>			<b>SUB-TURMAS</b>
Engenharia de Computação			C7
<b>PROFESSOR (ES) RESPONSÁVEL (EIS)</b>			<b>TITULAÇÃO</b>
Marcus Vinícius Midená Ramos			Doutor
<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 (recursos, materiais e procedimentos)</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 três notas.			



UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO  
CONSELHO UNIVERSITÁRIO

CONTEÚDO PROGRAMÁTICO	
	TEMAS ABORDADOS/DETALHAMENTO DA EMENTA
1	Apresentação e motivação Linguagens de alto e baixo-nível Linguagens fonte e objeto Compilação e interpretação Tipos de processadores Métodos para especificação de linguagens
2	Processadores de linguagens Diagramas-T Representação de processos
3	Compilador portátil Bootstrapping Desenvolvimento incremental de linguagens
4	Especificação de linguagens Revisão de gramáticas livres de contexto Sintaxe livre de contexto da linguagem mini-Triangle
5	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
6	Exemplo em mini-Triangle: programa-fonte, AST, AST decorada e programa-objeto Passos de compilação Um passo e vários passos: vantagens e desvantagens
7	Estrutura léxica Símbolos úteis e inúteis Tokens, tipo e valor Alfabeto1 -> linguagem1 = alfabeto2 -> linguagem2 Expressões regulares, BNF e EBNF
8	Substituição de não-terminais Fatoração à esquerda Eliminação de recursão à esquerda Starter Reduções e derivações Análise sintática descendente e ascendente
9	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
10	Exemplos e exercícios Gramática léxica Gramática sintática



**UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO**  
**CONSELHO UNIVERSITÁRIO**

11	Método recursivo descendente Aplicação do método recursivo descendente à construção de analisadores sintáticos
12	Aplicação do método recursivo descendente à construção de analisadores léxicos Aspectos de implementação
13	Desenvolvimento do projeto e esclarecimento de dúvidas
14	Montagem da AST Análise de contexto: identificação Tabela de identificação: atributos e métodos Linguagens com estrutura de bloco monolítica
15	Prova 1
16	Linguagens com estrutura de blocos planos Linguagens com estrutura de blocos aninhados Escopo estático x escopo dinâmico
17	Verificação de tipos Tipo estático x tipo dinâmico Implementação de analisadores de contexto Padrão de projeto Visitor
18	Representação de dados Tipos simples Tipos agregados heterogêneos
19	Tipos agregados homogêneos Verificação de índices Origem virtual Múltiplas dimensões
20	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
21	Alocação de memória - estática Alocação de memória - automática Estrutura do stack-frame (SB, LB, ST, L1, L2, ..., LD, LE, ER)
22	Passagem de parâmetros Retorno de valor de função Exemplo completo
23	Instruções CALL, RETURN, LOADA, LOADI, STOREI, PUSH e POP Passagem de parâmetros por endereço x por valor Exemplo completo
24	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
25	Exemplos de aplicação de funções e padrões de código Implementação dos métodos de geração de código



**UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO  
CONSELHO UNIVERSITÁRIO**

26	Padrão de código para declaração de procedimentos e funções, comandos for e switch; Organização da memória, pilha e heap.
27	Revisão Visitor Exemplo funcional completo com análise sintática, impressão da árvore de sintaxe, análise de contexto e geração de código (códigos fonte e objeto).
28	Orientação de projeto
29	Conclusões: Ciclo de vida de linguagens de programação Detecção e recuperação de erros Eficiência da execução Interpretores Geradores de compiladores
30	Prova 2

**REFERÊNCIAS BIBLIOGRÁFICAS**

- WATT D.A; BROWN, D.F. Programming Language Processors in Java - Compilers and Interpreters. Pearson Education, 2000.
- AHO, A.V. et al. Compiladores - Princípios, Técnicas e Ferramentas. 2ª ed. Pearson Education, 2007.
- PRICE, A.M.A.; TOSCANI, S.S. Implementação de Linguagens de Programação - Compiladores. Bookman, 2008.
- RICARTE I. Introdução à Compilação. Elsevier, 2008.
- DELAMARO, M.E. Como Construir um Compilador - Utilizando Ferramentas Java. Novatec, 2004.
- LOUDEN, K.C. Compiladores - Princípios e Práticas. Cengage Learning, 2004.
- GRÜNE, H.D. et al. Projeto Moderno de Compiladores - Implementação e Aplicações. Elsevier, 2001.

/ /	_____	/ /	_____
DATA	ASSINATURA DO PROFESSOR	HOMOLOGADO NO COLEGIADO	COORD. DO COLEGIADO