



Plano de Ensino

CAMPUS NOVA GAMELEIRA	
DISCIPLINA: Linguagens de Programação	CÓDIGO: G00LPRG0.01

Início: 01/2023

Carga Horária: Total: 60 horas-aula **Semanal:** 04 horas-aula **Créditos:** 04

Natureza: Teórica

Área de Formação - DCN: Profissionalizante

Competências/habilidades a serem desenvolvidas:

Compreender e aplicar os fundamentos de linguagens formais, os conceitos de linguagens de programação e as técnicas de construção de compiladores.

Departamento que oferta a disciplina: Departamento de Computação

Ementa:

Histórico e evolução de linguagens de programação. Conceitos de linguagens de programação: sistemas de linguagens, nomes, amarrações, escopo, tipos de dados, verificação de tipos, expressões e atribuições, estruturas de controle, subprogramas, tipos abstratos de dados, encapsulamento, polimorfismo, tratamento de exceções, localização e gerenciamento de memória. Paradigmas de linguagens de programação: procedurais, orientadas a objetos, orientadas a eventos, concorrentes, funcionais, lógicas.

Curso(s)	Período	Eixo	Obrigatória	Optativa
Engenharia de Computação	5º	Fundamentos da Computação	X	

Interdisciplinaridades:

Prerrequisitos
Algoritmos e Estruturas de Dados I Laboratório de Algoritmos e Estruturas de Dados I
Correquisitos

Objetivos: A disciplina deverá possibilitar ao estudante

1	Avaliar as vantagens e desvantagens de linguagens de programação.
2	Conhecer como linguagens são implementadas e suas implicações.
3	Aplicar os conceitos de linguagens e paradigmas na programação de computadores.

	Unidades de ensino	Carga horária horas-aula
1	Introdução: definições; sobre linguagens de programação; requisitos; critérios de avaliação; paradigmas (imperativo, funcional, orientado a objetos, lógico, entre outros); histórico.	2
2	Sintaxe e Semântica: representação de gramáticas (BNF e EBNF); árvores de derivação; construção, ambiguidade, precedência e	4

Plano de Ensino

	associatividade de gramáticas; árvores de sintaxe abstrata; estrutura léxica.	
3	Sistemas de linguagens: sequência clássica; variações da sequência clássica (interpretadores, máquinas virtuais, ligação tardia, perfilamento e compilação dinâmica, entre outros); depuradores; suporte em tempo de execução.	4
4	Programação funcional (SML básico): constantes; operadores; variáveis; tuplas; listas; funções; tipos; padrões.	4
5	Tipos: tipos primitivos e tipos compostos; anotações e inferência de tipos; verificação de tipos; equivalência de tipos.	4
6	Polimorfismo: sobrecarga; coerção; polimorfismo paramétrico e de subtipo; definições e classificações.	2
7	Programação funcional (SML avançado): funções como valores e anônimas; funções de ordem superior, funções pré-definidas (map, foldr, foldl); currying; enumerações; construtores (dados, tipos e tipos recursivos).	4
8	Nomes, amarrações e escopo: representação (variáveis, <i>labels</i> , subprogramas, parâmetros e outros); tempos de amarração (projeto, implementação, compilação, ligação, carga e execução); amarrações de tipos estáticas e dinâmicas; amarrações de armazenamento; escopo estático e dinâmico; tempo de vida.	4
9	Cálculo lambda: definições; notação; variáveis livres e ligadas; reduções (α -conversão, β -redução, η -conversão); combinadores (I, M, K, C, B).	4
10	Localização de variáveis: registros de ativação; alocações estáticas ou de pilha; funções encadeadas; funções como parâmetros.	2
11	Gerenciamento de memória: gerenciamento em tempo de execução; pilha e <i>heap</i> ; coletor automático de lixo.	2
12	Orientação a objetos: encapsulamento; herança; composição; polimorfismo <i>ad-hoc</i> (sobrecarga e coerção) e universal (inclusão e paramétrico).	4
13	Tratamento de exceções: tipos de exceções; tratamento e propagação; lançamento de exceções; exceções próprias.	4
14	Parâmetros: passagem de parâmetros; correspondência de parâmetros; implementação de parâmetros (por valor, resultado, valor-resultado; referência; nome; necessidade).	2
15	Programação concorrente: processos e <i>threads</i> ; sincronização de <i>threads</i> ; problema do produtor-consumidor; métodos de sincronização: semáforos e monitores.	4
16	Programação lógica (Prolog): termos, regras, operadores, listas, unificação, modelo de execução; cláusulas recursivas; computação numérica.	4
17	Modelos de custo: modelo de custo de listas, chamadas de funções, pesquisa em Prolog, arranjos e outros.	2



Plano de Ensino

18	Semântica formal: semântica natural; interpretadores em Prolog.	4
Total		60

Bibliografia Básica	
1	SEBESTA, R. W. Conceitos de linguagens de programação. 11. ed. Porto Alegre: Bookman, 2018.
2	WEBBER, A. B. Modern programming languages: a practical introduction. 2. ed. Franklin Beedle & Assoc, 2002.
3	VAREJAO, F. M. Linguagens de programação: conceitos e técnicas. 1. ed. Rio de Janeiro: Campus, 2004.

Bibliografia Complementar	
1	WATT, D. C.; FINDLAY, W. Programming language design concepts. Chichester: John Wiley & Sons, 2004.
2	LOUDEN, K. C.; LAMBERT K. A. Programming languages: principles and practices. 3 ed. Course Technology, 2011.
3	MITCHELL, J. C. Concepts in programming languages. 1. ed. Cambridge University Press, 2002.
4	GABBRIELLI, M.; MARTINI S. Programming languages: principles and paradigms. 1. ed. Springer, 2010.
5	SETTI, R. Programming Languages: concepts & constructs. 2.ed. Pearson, 2007.



PLANO DE ENSINO Nº 1559/2024 - CECOM (11.51.11)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 26/07/2024 16:25)

BRUNO ANDRE SANTOS

COORDENADOR

CECOM (11.51.11)

Matrícula: ##594#8

Visualize o documento original em <https://sig.cefetmg.br/documentos/> informando seu número: **1559**, ano: **2024**,
tipo: **PLANO DE ENSINO**, data de emissão: **26/07/2024** e o código de verificação: **31ff5007ca**