



PLANO DE ENSINO

DISCIPLINA: Programação de Computadores II	CÓDIGO: G08PROC2
---	-------------------------

Validade: Início: **02/2019** Término:
Carga Horária: **Total:** 30 horas/aula **Semanal:** 02 horas/aula **Créditos:** 02
Modalidade: Teórica
Classificação do Conteúdo pelas DCN: Básica

Ementa:

Conceitos de orientação a objetos: tipos abstratos de dados, objetos, classes, métodos, visibilidade, escopo, encapsulamento, associações de classes, estruturas todo-parte e generalização-especialização, interfaces; herança de interface e de classe, polimorfismo, sobrecarga, invocação de métodos; aplicações em uma linguagem de programação orientada a objetos; noções de modelagem de sistemas usando UML: diagrama de classes e de interação.

Cursos	Período	Eixo	Obrig.	Optativa
Engenharia Civil	3º	Computação e Matemática Aplicada	Sim	Não

Departamento/Coordenação: Departamento de Computação e Engenharia Civil

INTERDISCIPLINARIDADES

Pré-requisitos	Código
Programação de Computadores I	G08PROC1
Laboratório de Programação de Computadores I	G08LABP1
Co-requisitos	
Laboratório de Programação de Computadores II	G08LABP2
Disciplinas para as quais é pré-requisito	
Métodos de Elementos Finitos	G08METEF
Disciplinas para as quais é co-requisito	
Laboratório de Programação de Computadores II	G08LABP2

Objetivos: *A disciplina deverá possibilitar ao estudante*

1	conhecer e saber utilizar os conceitos de programação orientada a objetos.
2	projetar e implementar programas utilizando o paradigma de orientação a objetos

Unidades de Ensino	Carga-horária Horas/aula
1. INTRODUÇÃO AOS FUNDAMENTOS DE ORIENTAÇÃO A OBJETOS 1.1. Contexto histórico das linguagens de programação. 1.2. Paradigmas de programação.	02



PLANO DE ENSINO

1.3. Fundamentos das linguagens orientadas a objetos.	
2. APRESENTAÇÃO DE UMA LINGUAGEM ORIENTADA A OBJETOS 2.1. Histórico. 2.2. Fundamentos. 2.3. Sintaxe da linguagem. 2.4. Principais estruturas de controle. 2.5. Principais estruturas de dados.	10
3. INTRODUÇÃO A CLASSES E OBJETOS 3.1. Tipos abstratos de dados. 3.2. Classes e objetos. 3.3. Atributos e métodos. 3.4. Encapsulamento. 3.5. Construtores e destrutores.	04
4. HERANÇA 4.1. Visibilidade e escopo. 4.2. Herança Simples. 4.3. Herança Múltipla. 4.4. Hierarquia de classes.	04
5. POLIMORFISMO 5.1. Sobrecarga. 5.2. Sobreposição. 5.3. Uso do polimorfismo.	04
6. TRATAMENTO DE EXCEÇÕES 6.1. Conceitos básicos. 6.2. Hierarquia de exceções.	02
7. NOÇÕES DE MODELAGEM COM UML 7.1. Diagrama de classes. 7.2. Diagrama de interação.	04
TOTAL	30

Bibliografia Básica

1	MIZRAHI, V.V. Treinamento em linguagem C++: módulo 1. 2. ed. São Paulo: Pearson Prentice-Hall, 2006.
2	MIZRAHI, V.V. Treinamento em linguagem C++: módulo 2. 2. ed. São Paulo: Pearson Prentice Hall, 2006.
3	ZIVIANI, N. Projeto de algoritmos: com implementações em Java e C++. São Paulo: Thomson Learning, c2007.

Bibliografia Complementar

1	MATTHES, E. Curso intensivo de Python: uma introdução prática e baseada em projetos à programação. Novatec, 2016.
2	SLATKIN, B. Python eficaz: 59 maneiras de programar melhor em Python. Novatec, 2016.



MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DIRETORIA DE GRADUAÇÃO
COORDENAÇÃO DE GRADUAÇÃO EM ENGENHARIA CIVIL
UNIDADE VARGINHA

PLANO DE ENSINO

3	DEITEL, H.; DEITEL P. C++: como programar . 5. ed. São Paulo: Pearson Prentice Hall, 2006.
4	DROZDEK, A. Estrutura de dados e algoritmos em C++ . 2. ed. São Paulo: Cengage Learning, 2017.
5	HORSTMANN, C.S. Conceito de computação com o essencial de C++ . 3. ed. Porto Alegre: Bookman, 2005.

OBS.: Disciplina equalizada pela Resolução CGRAD 015/13

Varginha (MG), 18 de Novembro de 2019

Professor Daniel Guimarães do Lago

Coordenador Aellington Freire de Araújo