



FICHA DE COMPONENTE CURRICULAR

CÓDIGO: <u>FEELT31514</u>	COMPONENTE CURRICULAR: <u>PROGRAMAÇÃO ORIENTADA A OBJETOS</u>	
UNIDADE ACADÊMICA OFERTANTE: <u>FACULDADE DE ENGENHARIA ELÉTRICA</u>	SIGLA: <u>FEELT</u>	
CH TOTAL TEÓRICA: <u>30</u>	CH TOTAL PRÁTICA: <u>30</u>	CH TOTAL: <u>60</u>

OBJETIVOS

Ao final da disciplina o estudante será capaz de:

1. Desenvolver sistemas computacionais utilizando a técnica de orientação a objetos;
2. Resolver problemas de engenharia de computação e engenharia elétrica mapeados como uma solução orientada a objetos;
3. Criar uma classe de objetos referentes aos principais materiais, componentes e sistemas relacionados com engenharia elétrica e de computação.

EMENTA

Projeto orientado a objetos; classes; funções; alocação dinâmica de memória; herança; polimorfismo e templates.

DESCRIÇÃO DO PROGRAMA

1. Introdução à orientação a objetos

- 1.1. Paradigmas de linguagens de programação

- 1.2. Classes e objetos
- 1.3. Modelo de um objeto
- 1.4. Relações entre objetos
- 1.5. Notação de Booch para objetos
- 1.6. Estudo de caso na Engenharia: simulação de áreas e volumes

2. Classes

- 2.1. Atributos de uma classe
- 2.2. Operações de uma classe
- 2.3. Construtores e destrutores
- 2.4. Criação de classes de componentes elétricos
- 2.5. Estudo de caso na Engenharia: modelagem UML de classes de componentes elétricos

3. Funções

- 3.1. Funções do tipo *inline*
- 3.2. Sobrecarga de funções
- 3.3. Classes amigas (*friend classes*)
- 3.4. Estudo de caso na Engenharia: Funções de filtros de dados

4. Alocação dinâmica de memória e orientação a objetos

- 4.1. Bibliotecas para alocação dinâmica de memória
- 4.2. Coleta de lixo
- 4.3. Alocação dinâmica de construtores

5. Herança

- 5.1. Propósitos da herança
- 5.2. Tipos de herança
- 5.3. Construção de objetos por herança
- 5.4. Destrução de objetos por herança
- 5.5. Heranças múltiplas
- 5.6. Estudo de caso na Engenharia: Aplicação de herança na modelagem e relação de componentes de circuitos e instalações elétricas

6. Polimorfismo

- 6.1. Definição de polimorfismo
- 6.2. Funções virtuais
- 6.3. Classes abstratas
- 6.4. Polimorfismo e gerenciamento de classes
- 6.5. Dynamic Casting

- 6.6. Identificação de tipos em tempo de execução
- 6.7. Estudo de caso na Engenharia: Simulação de circuitos elétricos utilizando polimorfismo

7. Templates

- 7.1. Definição
- 7.2. Funções e classes
- 7.3. Herança

BIBLIOGRAFIA

BIBLIOGRAFIA BÁSICA:

1. MIZRAHI, V. V., Treinamento em Linguagem C++. Módulo 2, 2ª. Edição, Pearson Prentice Hall, São Paulo, 2006.
2. KEOGH, J.; GRANNINI, M., OOP Desmistificado - Programação Orientada A Objetos. ALTA BOOKS, Rio de Janeiro, 2005
3. HEMRAJANI, A., Desenvolvimento Ágil em Java. Pearson Prentice Hall, São Paulo, 2007

BIBLIOGRAFIA COMPLEMENTAR:

1. ETTER, D. M. and INGBER, J. A., Engineering Problem Solving with C++. 2nd Edition, Prentice Hall, 2008.
2. MOTA, A. A., Programação Orientada A Objetos com C++, Relativa Editora, São Paulo, 2002.
3. FREEMAN, Eric; FREEMAN, Elisabeth. Use a cabeca! padrões de projeto. Rio de Janeiro: Atlas Books, 2005.
4. BOOCHE, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. Rio de Janeiro: Campus, 2000.
5. DEITEL, Harvey M. Java: Como Programar, 8ed. São Paulo: Prentice Hall, 2010.

APROVAÇÃO

____ / ____ / _____

____ / ____ / _____

Carimbo e assinatura do
Coordenador do curso

Carimbo e assinatura do
Diretor da Unidade Acadêmica