



UNIVERSIDADE ESTADUAL DE MARINGÁ
PRÓ-REITORIA DE ENSINO

PROGRAMA DE DISCIPLINA

Curso:	Bacharelado em Ciência da Computação	Campus:	Maringá - Sede
Departamento:	Departamento de Informática		
Centro:	Centro de Tecnologia		
COMPONENTE CURRICULAR			
Nome: Paradigma de Programação Imperativa e Orientada a Objetos			Código: 6888
Carga Horária: 68	Periodicidade: Semestral	Ano de Implantação: 2012	
1. EMENTA			
Classes de linguagens de programação. Avaliação de linguagens de programação. Paradigma imperativo:. Paradigma orientado a objetos.			
2. OBJETIVOS			
<ul style="list-style-type: none">• Apresentar a fundamentação teórica sobre o paradigma imperativo, enfatizando os conceitos sobre: nomes, escopos e ligações; fluxo de controle; tipos de dados; subrotinas e abstração de controle.• Apresentar a fundamentação teórica sobre o paradigma orientado a objetos, enfatizando os conceitos sobre: tipos abstratos de dados; herança e polimorfismo.• Consolidar conceitos e desenvolver habilidades de programação nos paradigmas imperativo e orientado a objetos.• Conduzir a implementação de programas de porte médio utilizando os paradigmas imperativo e orientado a objetos.			

3. CONTEÚDO PROGRAMÁTICO
<p><u>Teoria:</u></p> <ol style="list-style-type: none">1. Introdução<ol style="list-style-type: none">1.1. Domínios de Programação1.2. Classes de Linguagens1.3. Critérios de Avaliação1.4. Métodos de Implementação2. Nomes, Escopos e Ligações<ol style="list-style-type: none">2.1. Tempo de Ligação2.2. Tempo de Vida e Gerência de Armazenamento2.3. Regras de Escopo2.4. Nomes e Escopo2.5. Ligação de Ambientes de Referenciamento

- 3. Fluxo de Controle
 - 3.1. Avaliação de Expressão
 - 3.2. Controle de Fluxo Estruturado e Não Estruturado
 - 3.3. Sequenciamento
 - 3.4. Seleção
 - 3.5. Iteração
 - 3.6. Recursão

- 4. Tipos de Dados
 - 4.1. Sistemas de Tipo
 - 4.2. Checagem de Tipos
 - 4.3. Registro e União
 - 4.4. Arrays
 - 4.5. Strings
 - 4.6. Conjuntos
 - 4.7. Ponteiros e Tipos Recursivos

- 5. Subrotinas e Abstração de Controle
 - 5.1. Passagem de Parâmetros
 - 5.2. Subrotinas Genéricas e Módulos
 - 5.3. Gerenciamento de Exceções
 - 5.4. Corotinas
 - 5.5. Eventos

- 6. Tipos de Dados Abstratos e Encapsulamento
 - 6.1. Abstração de Dados
 - 6.2. Questões de Projeto em Abstração de Dados
 - 6.3. Tipos de Dados Abstratos Parametrizados
 - 6.4. Encapsulamento

- 7. Abstração de Dados e Orientação a Objetos
 - 7.1. Programação Orientada a Objetos
 - 7.2. Encapsulamento e Herança
 - 7.3. Inicialização e Finalização
 - 7.4. Ligação Dinâmica
 - 7.5. Herança Múltipla

Prática:

Aulas práticas em conformidade com o conteúdo teórico.

4. REFERÊNCIAS

4.1- Básicas (Disponibilizadas na Biblioteca ou aquisições recomendadas)

- SCOTT, Michael L. Programming Language Pragmatics. 3Ed: Morgan Kaufmann, 2009,
- SEBESTA, Robert W. Concepts of Programming Languages. 9Ed: Addison Wesley, 2010.
- FRIEDMAN, Daniel P. and WAND, Mitchel. Essentials of Programming Languages. 3Ed: The MIT Press, 2008.
- TUCKER, Allen B.; NOONAN, Robert E. Linguagens de Programação: Princípios Básicos. 2Ed: Mc Graw Hill, 2009.

TURBAK, Franklyn A., GIFFORD, David. K. Design Concepts in Programming Languages. MIT PRESS, 2008.

WATT, David A. Programming Language Design Concepts. IE-Wiley, 2004.

STEVENSON, D. E. Programming Language Fundamentals by Example. CRC Press, 2006.

GHEZZI, Carlo. Programming Language Concepts. 3Ed: Wiley, 1997.

HOROWITZ, Ellis. Fundamentals of Programming Languages. 2Ed: Computer Science Press, 1984

VAREJÃO, F. Linguagens de Programação. Campus, 2004.

MELO, Ana C. V., SILVA, Flavio S. C. Princípios de Linguagens de Programação. Edgard Blucher, 2003.

WEBBER, Adam B. Modern Programming Languages: A Practical Introduction. Franklin Beedle & Associates, 2002.

4.2- Complementares

Periódicos e anais de conferências.

APROVAÇÃO DO DEPARTAMENTO

APROVAÇÃO DO CONSELHO
ACADÊMICO