



**UNIVERSIDADE ESTADUAL DE MARINGÁ  
PRÓ-REITORIA DE ENSINO**

**PROGRAMA DE DISCIPLINA**

Curso:	<b>Ciência da Computação</b>		
Departamento:	Informática – DIN		
Centro:	Centro de Tecnologia - CTC		
<b>COMPONENTE CURRICULAR</b>			
Nome: <b>Linguagens de Programação</b>			Código: <b>1028</b>
Carga Horária: <b>68 horas/aula</b>	Periodicidade: <b>Anual</b>	Ano de Implantação: <b>2009</b>	
<b>1. EMENTA</b>			
Estudo de paradigmas e avaliação de linguagens de programação.			
<b>2. OBJETIVOS</b>			
<ul style="list-style-type: none"><li>• Estudar as características peculiares de algumas linguagens de programação.</li><li>• Realizar estudo comparativo de linguagens de programação.</li></ul>			
<b>3. CONTEÚDO PROGRAMÁTICO</b>			
<ol style="list-style-type: none"><li>1. Aspectos preliminares<ol style="list-style-type: none"><li>1.1. Domínios de programação</li><li>1.2. Critérios para avaliar uma linguagem de programação</li><li>1.3. Influências no projeto da linguagem</li><li>1.4. Categorias de linguagens</li><li>1.5. Métodos de implementação</li></ol></li><li>2. Nomes e vinculações<ol style="list-style-type: none"><li>2.1. Nomes</li><li>2.2. Variáveis</li><li>2.3. Vinculação</li><li>2.4. Verificação de tipos, tipificação forte e compatibilidade de tipos</li><li>2.5. O escopo de um nome e ambientes de referência</li></ol></li><li>3. Tipos de dados<ol style="list-style-type: none"><li>3.1. Tipos primitivos, strings e definidos pelo usuário</li><li>3.2. Arrays</li><li>3.3. Registros</li><li>3.4. Uniões</li><li>3.5. Ponteiros e Referências</li></ol></li><li>4. Expressões e instruções de atribuição<ol style="list-style-type: none"><li>4.1. Expressões aritméticas</li><li>4.2. Operadores sobrecarregados</li><li>4.3. Casts, conversões e coerções de tipos</li></ol></li></ol>			

- 4.4. Expressões relacionais e booleanas
- 4.5. Avaliação em circuito-curto
- 4.6. Instruções de atribuição
- 4.7. Atribuição de modo misto
  
- 5. Estruturas de controle
  - 5.1. Estruturas de seleção
  - 5.2. Estruturas iterativas
  - 5.3. Saltos
  - 5.4. Comandos guardados
  
- 6. Subprogramas e parâmetros
  - 6.1. Fundamentos dos subprogramas
  - 6.2. Questões de projeto referentes aos subprogramas
  - 6.3. Ambientes de referência locais
  - 6.4. Métodos de passagem de parâmetros
  - 6.5. Parâmetros que são nomes de subprogramas
  - 6.6. Subprogramas sobrecarregados
  - 6.7. Subprogramas genéricos
  - 6.8. Questões de projeto referentes a funções
  - 6.9. Operadores sobrecarregados definidos pelo usuário
  
- 7. Implementando subprogramas
  - 7.1. A semântica geral das chamadas e dos retornos
  - 7.2. Implementando subprogramas simples
  - 7.3. Implementando subprogramas com variáveis locais dinâmicas na pilha
  - 7.4. Implementando subprogramas aninhados
  - 7.5. Blocos
  - 7.6. Implementando escopo dinâmico
  
- 8. Tipos de Dados Abstratos e Construções Encapsuladas
  - 8.1. O conceito de abstração
  - 8.2. Introdução à abstração de dados
  - 8.3. Questões de projeto
  - 8.4. Exemplos de linguagens
  - 8.5. Tipos de dados abstratos parametrizados
  - 8.6. Encapsulamento
  
- 9. Suporte a Programação Orientada a Objeto
  - 9.1. Programação orientada a objeto
  - 9.2. Questões de projeto
  - 9.3. Suporte para programação orientada a objeto em Smalltalk, C++, Java, C#, Ada 95 e Ruby
  - 9.4. O Modelo de objetos de JavaScript
  - 9.5. Implementação de construções orientadas a objeto
  
- 10. Concorrência
  - 10.1. Concorrência no nível do subprograma
  - 10.2. Semáforos
  - 10.3. Monitores
  - 10.4. Passagem de mensagem
  - 10.5. Suporte a concorrência em Ada
  - 10.6. Threads em Java
  - 10.7. Threads em C#
  - 10.8. Concorrência em nível de instruções

## 11. Manipulação de Exceções e Manipulação de Eventos

- 11.1. Introdução à manipulação de exceções
- 11.2. Manipulação de exceções em Ada, C++ e Java
- 11.3. Introdução à manipulação de eventos
- 11.4. Manipulação de eventos em Java

## 12. Linguagens de Programação Funcional

- 12.1. Fundamentos das linguagens de programação funcionais
- 12.2. A primeira linguagem de programação funcional: LISP
- 12.3. Exemplos de linguagens funcionais: Scheme, Common LISP, ML e Haskell
- 12.4. Aplicações de linguagens funcionais
- 12.5. Uma comparação entre as linguagens funcionais e imperativas

## 13. Linguagens de Programação Lógicas

- 13.1. Uma breve introdução ao cálculo de predicados
- 13.2. Uma visão geral de programação lógica
- 13.3. As origens da linguagem Prolog
- 13.4. Os elementos básicos da linguagem Prolog
- 13.5. Aplicações da programação lógica

## 4. REFERÊNCIAS

### 4.1- Básicas (Disponibilizadas na Biblioteca ou aquisições recomendadas)

SEBESTA, Robert W. Concepts of Programming Languages. 8<sup>th</sup> Edition, Addison Wesley, 2008.

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

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

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

FRIEDMAN, Daniel P., WAND, Mitchell. Essencials of Programming Languages. 3<sup>th</sup> Edition, The MIT Press, 2008.

TUCKER, Allen B. Programming Languages: Principles and Paradigms. McGraw-Hill Higher Education, 2007.

SCOTT, Michael L. Programming Language Pragmatics. 2<sup>th</sup> Edition, Morgan Kaufmann, 2005.

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

### 4.2- Complementares

