

<b>DISCIPLINA:</b> Programação de Computadores II	<b>CÓDIGO:</b> 2ECOM.007
---	--------------------------

**VALIDADE:** a partir de agosto de 2010

**TÉRMINO:**

**Carga Horária:** Total: 30 horas-aula Semanal: 02 aulas 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.

Curso (s)	Período	Eixo	Natureza
Engenharia de Computação	2	Fundamentos de Engenharia de Computação	Obrigatória
Engenharia Elétrica	2	Computação e Matemática Aplicada	Obrigatória
Engenharia Mecânica	2	Computação e Matemática Aplicada	Obrigatória
Engenharia de Produção Civil	3	Computação e Matemática Aplicada	Obrigatória
Engenharia Mecatrônica	2	Programação de Computadores e Computação Aplicada	Obrigatória
Engenharia de Controle e Automação	2	Computação e Matemática Aplicada	Obrigatória
Engenharia de Materiais	3	Linguagem de Programação e Expressão Gráfica	Obrigatória
Engenharia de Automação Industrial	2	Computação e Matemática Aplicada	Obrigatória

**Departamento/Coordenação:** Departamento de Computação (DECOM)

## INTERDISCIPLINARIEDADES

<b>Pré-requisitos</b>
- Programação de Computadores I. - Laboratório de Programação de Computadores I.
<b>Co-requisitos</b>
- Laboratório de Programação de Computadores II.
<b>Disciplinas para as quais é pré-requisito</b>
- Algoritmos e Estruturas de Dados I (Engenharia de Computação) - Linguagens de Programação (Eng. Mecatrônica) - Análise e Projeto de Algoritmos (Eng. Mecatrônica) - Tópicos Especiais em Programação de Computadores e Computação Aplicada (Eng. Mecatrônica) - Métodos Numéricos Computacionais (Eng. de Computação; Eng. Elétrica, Eng. Mecânica, Eng. de Produção Civil, Eng. Mecatrônica, Eng. de Controle e Automação, Eng. de Materiais e Eng. de Automação Industrial)
<b>Disciplinas para as quais é co-requisito</b>
- Lab. de Programação de Computadores II
<b>Transdisciplinariedade (inter-relações desejáveis)</b>
-----

<b>Objetivos:</b> <i>A disciplina deverá possibilitar ao estudante</i>
--

- |   |
|---|
| <ul style="list-style-type: none"><li>- Conhecer e saber utilizar os conceitos de programação orientada a objetos.</li><li>- Projetar e implementar programas utilizando o paradigma de orientação a objetos.</li></ul> |
|---|

Unidade de ensino		Carga-horária Horas-aula
1	<p>Introdução a C++ e aos fundamentos de orientação a objetos</p> <ul style="list-style-type: none"> <li>Contexto histórico das linguagens de programação</li> <li>Paradigmas de programação</li> <li>Fundamentos das linguagens orientadas a objetos</li> </ul>	2
2	<p>Programação em C++</p> <ul style="list-style-type: none"> <li>Conceitos básicos em C++</li> <li>Constantes, tipos, operadores</li> <li>Estruturas de controle</li> <li>Arrays e apontadores</li> <li>Alocação dinâmica de memória</li> </ul>	4
3	<p>Funções, sobrecarga e namespaces</p> <ul style="list-style-type: none"> <li>Namespaces e escopo</li> <li>Funções: declaração e definição</li> <li>Funções recursivas</li> <li>Funções inline</li> <li>Sobrecarga de funções</li> <li>Passagem de parâmetro por cópia e por referência</li> </ul>	4
4	<p>Introdução a classes e objetos</p> <ul style="list-style-type: none"> <li>Abstração</li> <li>Encapsulamento</li> <li>Tipos abstratos de dados</li> <li>Classes: definição e uso</li> <li>Objetos: instâncias de classes</li> </ul>	2
5	<p>Funções implícitas e sobrecarga de operadores</p> <ul style="list-style-type: none"> <li>Funções construtoras e destruidoras</li> <li>Sobrecarga de operadores</li> <li>Apontador this</li> </ul>	2
6	<p>Tipos específicos de funções e amizade</p> <ul style="list-style-type: none"> <li>Funções constantes</li> <li>Funções estáticas</li> <li>Funções amigas</li> <li>Classes amigas</li> </ul>	2
7	<p>Herança</p> <ul style="list-style-type: none"> <li>Visibilidade, escopo, especificadores de acesso</li> <li>Classes base e derivada</li> <li>Herança de construtores e destruidores</li> <li>Hierarquia de classes</li> <li>Herança múltipla</li> </ul>	6
8	<p>Polimorfismo de tipos</p> <ul style="list-style-type: none"> <li>Funções virtuais e virtuais puras</li> </ul>	

Plano de Ensino

	<ul style="list-style-type: none"><li>• Classes abstratas</li><li>• Tipos polimórficos</li><li>• Uso do polimorfismo</li></ul>	4
9	Outros tópicos <ul style="list-style-type: none"><li>• Formatação de saída</li><li>• Templates ou gabaritos</li><li>• Tratamento de exceção</li></ul>	2
10	Noções de modelagem com UML <ul style="list-style-type: none"><li>• Diagrama de classes</li><li>• Diagrama de interação</li></ul>	2
<b>Total</b>		30

**Bibliografia Básica**

LAFORE, R. **Object-oriented programming in C++**. 4th ed. Indianapolis: Sams, 2002.

**Bibliografia Complementar**

DEITEL, H. M.; DEITEL, P., J. **Java: como programar**. 8. ed. São Paulo: Pearson Prentice-Hall, 2010.

LIPMAN, S. B.; LAJOIE, J. **A C++ primer**. 4th ed. New York: Addison-Wesley, 2005.