



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**  
**FACULDADE DE COMPUTAÇÃO**  
**COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**FICHA DE DISCIPLINA**

**DISCIPLINA: PROGRAMAÇÃO ORIENTADA A OBJETOS 1**

<b>CÓDIGO: GBC035</b>		<b>UNIDADE ACADÊMICA:</b> FACULDADE DE COMPUTAÇÃO		
<b>PERÍODO/SÉRIE: - 3º. Período</b>		<b>CH TOTAL TEÓRICA:</b>	<b>CH TOTAL PRÁTICA:</b>	<b>CH TOTAL:</b>
<b>OBRIGATORIA: ( X )</b>	<b>OPTATIVA: ( )</b>	30	30	60
<b>NÚCLEO DE FORMAÇÃO: Básico</b>				
<b>PRÉ-REQUISITOS: NÃO HÁ</b>		<b>CÓ-REQUISITOS: NÃO HÁ</b>		

**OBJETIVOS**

Ao final do curso o aluno será capaz de:

1. Analisar problemas computacionais e propor soluções utilizando conceitos de programação orientada a objetos, como classes, objetos, herança e polimorfismo
2. Desenvolver programas em uma linguagem de programação orientada a objetos.

**EMENTA**

Introdução dos conceitos fundamentais de programação orientada a objetos, como classes, objetos, encapsulamento, herança e polimorfismo. Conhecimento dos membros que tipicamente compõem classes: construtores, destrutores, variáveis e métodos. Entendimento e aplicação dos conceitos de orientação a objetos em linguagens de programação que suportem tal paradigma. Desenvolvimento de sistemas usando programação orientada a objetos.

## DESCRIÇÃO DO PROGRAMA



1. Introdução à programação orientada a objetos
  - 1.1. Histórico da programação orientada a objetos
  - 1.2. Programação procedimental versus programação orientada a objetos
2. Conceitos fundamentais de programação orientada a objetos
  - 2.1. Classes concretas, classes abstratas e interfaces
  - 2.2. Objetos, atributos, métodos, construtores e destrutores
  - 2.3. Membros de classe e membros de objetos
  - 2.4. Métodos concretos e métodos abstratos
  - 2.5. Pacotes, visibilidade e encapsulamento
  - 2.6. Generalização, especialização e herança
  - 2.7. Polimorfismo
  - 2.8. Ligação estática e ligação dinâmica
3. Uso de bibliotecas em linguagens orientadas a objetos
  - 3.1. Tratamento de exceções
  - 3.2. Uso de API básica (e.g., *java.lang* e *java.util*)
  - 3.3. Manipulação de arquivos (e.g., *java.io*)
  - 3.4. Interfaces gráficas simples (e.g., *java.awt* e *javax.swing*)
  - 3.5. Acessos ao banco de dados
4. Desenvolvimento de um sistema orientado a objetos
  - 4.1. Implementação usando programação orientada a objetos
  - 4.2. Documentação do sistema desenvolvido

## BIBLIOGRAFIA

### Básica:

BOOCH, G. Object-Oriented Analysis and Design with Applications, 3a Edição. Addison-Wesley, 2007.  
BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. Rio de Janeiro: Campus, 2000.  
DEITEL, H. M.; DEITEL P. J. Java: Como Programar, 6a. Edição. Pearson, 2005. (Livro Texto)

### Complementar:

ECKEL, B. Thinking in Java, 2a Edição. Prentice Hall, 2000.  
FLANAGAN, D. Java in a Nutshell, 5a. Edição. O'Reilly Assoc. Inc. 2005.  
FOWLER, M. UML Essencial, 2a Edição. Bookmann, 2000.  
HORSTMANN, C.; CORNELL, G. Core Java 2 - Fundamentals, 7a. Edição. Prentice Hall, 2004.  
LARMAN, C. Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientado a Objetos. Porto Alegre: Bookmann, 2001.  
RUMBAUGH, J.; BLAHA, M. Modelagem e Projetos Baseados em Objetos com UML 2, 1a Edição. Editora Campus, 2006.

## APROVAÇÃO

Carimbo e assinatura do Coordenador do curso  
Universidade Federal de Uberlândia  
Prof. Inédio Cordeiro  
Coordenador do Curso de Ciência da Computação  
Portaria R nº 713/08

Carimbo e assinatura do Diretor da  
Unidade Acadêmica  
Universidade Federal de Uberlândia  
Prof. Jamil Saleem Barbar  
Diretor da Faculdade de Computação  
Portaria R nº 672/07