

---

**CURSO:** ENGENHARIA DE SOFTWARE

**DISCIPLINA:** Orientação a Objetos

**CÓDIGO**

**195341**

**TURMA:** B e D

**CRÉDITOS:**

**04**

**SEMESTRE:** 2020 / 1

**PROFESSOR:** Daniel de Paula Porto

---

## PLANO DE ENSINO

---

### 1. OBJETIVOS DA DISCIPLINA

---

Abordar as principais características do paradigma de orientação a objetos, tais como: classes, objetos, atributos, métodos, herança, polimorfismo e encapsulamento. Introduzir os conceitos fundamentais necessários para o desenvolvimento de programas de computador baseados neste paradigma. Introduzir os conceitos de análise e concepção orientadas a objetos (modelagem orientada a objetos), com uso inicial da notação UML (*Unified Modeling Language*).

---

### 2. EMENTA

---

1. Problemas no desenvolvimento de software;
  2. Introdução às boas práticas de programação;
  3. Introdução à programação orientada a objetos;
  4. Projetos e implementação de sistema orientado a objetos.
- 

### 3. HORÁRIO DAS AULAS E DE ATENDIMENTO

---

**Aulas teóricas / práticas:** Turma B (Seg 08h, Qua 08h) na sala I7 / Turma D (Seg 16h, Qua 16h) na sala I6.

**Horário de atendimento:** Segunda-feira, das 10h às 12h na sala 35.

**Atendimento virtual:** Fórum de Dúvidas disponível no ambiente Aprender (<http://aprender.unb.br>)

---

### 4. CONTEÚDO PROGRAMÁTICO

---

01. Introdução a Orientação a Objetos
  02. Classes e objetos
  03. Introdução a Linguagem JAVA
  04. Encapsulamento
  05. Herança
  06. Polimorfismo
  07. Desenvolvimento de aplicações orientadas a objetos
- 

### 5. METODOLOGIA

---

Serão ministradas aulas expositivas e práticas em laboratório pelo professor e auxiliadas por monitores (se disponíveis). O objetivo das práticas é permitir que os estudantes desenvolvam habilidades essenciais para a programação orientada a objetos, além de fixar os conceitos abordados nas aulas teóricas e das compreensões lógicas e organizacionais ao emprego adequado da tecnologia de orientação a objetos.

Com o intuito de estabelecer a aprendizagem nessa disciplina, as aulas serão complementadas com atividades de exercícios, trabalhos e demandas extraclasse, disponibilizadas por meio do Ambiente Virtual de Aprendizagem

(MOODLE), além do uso de um Ambiente de Desenvolvimento Integrado (Eclipse – IDE - *Integrated Development Environment*).

## 6. CRITÉRIOS DE AVALIAÇÃO

A avaliação dos alunos na disciplina será composta por **2 provas (P1 e P2), atividades extraclasse (E) e um trabalho (T)**. Cada prova será realizada em laboratório e engloba toda a matéria apresentada anteriormente (conteúdo cumulativo).

A Nota Final na disciplina será calculada da seguinte forma:

$$NF = (P1 * 0,20) + (P2 * 0,40) + (E * 0,10) + (T * 0,30)$$

Para ser aprovado na disciplina o aluno precisa:

- Ter 75% de presença nas aulas;
- Atingir Nota Final (NF) maior ou igual a 5,0.

As atividades extraclasse envolvem a realização de exercícios de programação em ambiente virtual.

Os estudantes poderão participar de qualquer atividade avaliativa desde que cheguem no local da aplicação até dez minutos depois do início da mesma, sendo seu prazo limite para encerramento devidamente informado pelo professor, após o início de cada prova (atividade avaliativa). As atividades extraclasse envolvem a realização de exercícios, sua implementação e a entrega em ambiente virtual no padrão lecionado e exigido pelo professor. Só assim serão considerados realizados a contento aos objetivos da disciplina. As atividades só são consideradas realizadas quando forem entregues, exatamente, no ambiente em que foram solicitadas, respeitando seus prazos limites de elaboração e entrega definidos e divulgados pelo professor, além dos padrões adotados pela disciplina.

A impossibilidade de comparecimento de qualquer estudante a uma das atividades avaliativas deverá ser comprovada por documento oficial e original (Atestado Médico, Declaração de Serviço Militar, etc.) indicando o motivo justificável perante a lei de sua ausência, em que será possível que este estudante realize uma Prova de Reposição (**PR**) na data previamente agendada para o final deste semestre letivo (ver Cronograma de Atividades). Todo o conteúdo lecionado por esta disciplina fará parte desta avaliação (**PR**).

## 7. CRONOGRAMA DE ATIVIDADES

SEMANA	DATA	ATIVIDADES / CONTEÚDOS
1ª	09/03	Apresentação do Plano de Ensino Introdução ao Paradigma de Programação Orientado a Objeto com Linguagem Java
	11/03	Ambiente de Desenvolvimento Orientação sobre a instalação do ambiente de desenvolvimento (Eclipse e Java) Atividade de Programação e preparação do ambiente para a disciplina (Eclipse e Java) Tipos Primitivos e Operadores Convenções Controle de Fluxo (condicional e repetição)
2ª	16/03	Introdução a <i>Arrays</i> Exercícios de fixação envolvendo Controle de Fluxo
	18/03	Introdução a <i>String</i> Exercícios de fixação envolvendo Controle de Fluxo
3ª	23/03	Classes e Objetos Declarações de classes e métodos Instanciação e referência de objetos
	25/03	<i>Arrays</i> com mais dimensões e <i>Strings</i> mutáveis Introdução a Exceções (tratamento)
4ª	30/03	Classes e Objetos (continuação) Modelagem e Notação UML (Casos de Uso)
	01/04	Encapsulamento
5ª	06/04	Pacotes (na organização da implementação) Exercícios de Fixação
	08/04	Exercícios de Revisão

6ª	13/04	<b>PRIMEIRA AVALIAÇÃO (P1)</b>
	15/04	Correção da Avaliação P1
7ª	20/04	Coleções - Vector <i>Generics</i> (parametrização)
	22/04	Coleções <i>ArrayList</i>
8ª	27/04	Janela de Diálogo
	29/04	Herança (início)
9ª	04/05	Herança (continuação)
	06/05	Polimorfismo Notação UML (Diagrama de Classe)
10ª	11/05	Classe Abstrata
	13/05	Interface
11ª	18/05	Coleções ( <i>Collection</i> ) Exceções (propagação e criação)
	20/05	Programação com interface gráfica (GUI)
12ª	25/05	Programação com Eventos Padrões de Projeto (MVC)
	27/05	Armazenamento Secundário (BD) Padrão DAO
13ª	01/06	Exercícios de Fixação
	03/06	Controle de versão (Git)
14ª	08/06	Introdução ao Desenvolvimento na Web (HTML e HTTP)
	10/06	Introdução ao Desenvolvimento Java na Web (JSP e <i>Servlets</i> )
15ª	15/06	<b>SEGUNDA AVALIAÇÃO (P2)</b>
	17/06	Correção da Avaliação P2
16ª	22/06	Acompanhamento com apresentação dos Trabalhos Finais (T)
	24/06	Apresentação e Entrega dos Trabalhos Finais (T)
17ª	29/06	Apresentação e Entrega dos Trabalhos Finais (T)
	01/07	<b>PROVA DE REPOSIÇÃO (PR)</b>
18ª	06/07	Divulgação dos Resultados Finais
	08/07	Possível aula de reposição prevista no calendário da UnB (se for necessário)

## 8. BIBLIOGRAFIA

### BÁSICA

STROUSTRUP, B. The C++ Programming Language, 4th Edition, Addison-Wesley Professional, 2013.  
 HORSTMANN, C. S.; CORNELL, G. Core Java, Volume I - Fundamentals, 8th Edition, Prentice Hall, 2016.  
 (eBrary) LANO, K. UML 2 Semantics and Applications, Wiley, 2009.

### COMPLEMENTAR

IAN D. CHIVERS, An Introduction to C++ and Object Oriented Programming, Springer, 2001.  
 (eBrary) BARCLAY, K.; SAVAGE, J. Object-Oriented Design with UML and Java, Elsevier, 2003.  
 (eBrary) SHAUGHNESSY, P. Ruby Under a Microscope: An Illustrated Guide to Ruby Internals, No Starch Press, 2013.  
 (eBrary) PHILLIPS, D. Python 3 Object Oriented Programming, Packt Publishing Ltd, 2010.  
 DEITEL, H. M., DEITEL, P. J. Java: Como Programar, 8 ed. Pearson do Brasil, 2010.  
 HORSTMANN, C. S. e CORNELL, G. Core Java2, volume 2, Makron Books, 2001.

## 9. OBSERVAÇÕES

- Conforme a conveniência, a data das avaliações e a distribuição dos conteúdos podem ser alteradas, desde que com prévia comunicação e anuência dos estudantes.

2. O estudante que faltar a uma das atividades avaliativas terá direito a fazer uma Prova de Reposição (PR) prevista no Cronograma de Atividades, desde que apresente atestado/declaração que justifique a ausência e seja coerente aos aspectos legais que aceitam tal ausência como justificada no Brasil.
3. Caso haja necessidade de reposição de aula no decorrer desse semestre letivo, as mesmas serão ministradas em datas e horários não previstos neste cronograma (item 7), sendo estas, antecipadamente, combinadas entre o professor e os estudantes desta disciplina/turma.