



## PLANO DE ENSINO

<b>CURSO</b>	Engenharia de Computação	<b>MATRIZ</b>	21
--------------	--------------------------	---------------	----

<b>FUNDAMENTAÇÃO LEGAL</b>	Resolução nº 024/2014 do COGEP em 24 de abril de 2014.
----------------------------	--

DISCIPLINA/UNIDADE CURRICULAR	CÓDIGO	PERÍODO	CARGA HORÁRIA (aulas)					
			AT	AP	APS	AD	APCC	Total
Fundamentos de Orientação a Objetos	CP44E	4º	34	34	04	00	00	72

AT: Atividades Teóricas, AP: Atividades Práticas, APS: Atividades Práticas Supervisionadas, AD: Atividades a Distância, APCC: Atividades Práticas como Componente Curricular.

<b>PRÉ-REQUISITO</b>	CP42F.
<b>EQUIVALÊNCIA</b>	ET43G, TSI32A.

### OBJETIVOS

A disciplina objetiva propiciar os conhecimentos necessários ao discente para desenvolver as competências de conhecer os conceitos básicos do Paradigma Orientado a Objetos (OO), diferenciar técnicas estruturadas da abordagem OO no desenvolvimento de software e aplicar os conceitos de OO na implementação de programas utilizando ambientes integrados de desenvolvimento e linguagens OO.

### EMENTA

Paradigma orientado a objetos (fundamentos, aplicações, benefícios); Conceitos: classes, objetos, atributos, métodos, mensagens, encapsulamento, herança, polimorfismo, construtores e sobrecarga de métodos; Classes abstratas e interfaces; Arrays; Coleções. Tratamento de exceções.

### CONTEÚDO PROGRAMÁTICO

ITEM	EMENTA	CONTEÚDO
1	Paradigma orientado a objetos	1.1 Fundamentos; 1.2 Aplicações; 1.3 Benefícios.
2	Conceitos de orientação a objetos	2.1 Conceitos 2.1.1 Objetos; 2.1.2 Classes; 2.1.3 Atributos; 2.1.4 Métodos; 2.1.5 Mensagens; 2.1.6 Encapsulamento; 2.1.6.1 Ocultamento; 2.1.7 Herança; 2.1.7.1 Generalização e especialização; 2.1.8 Polimorfismo; 2.1.9 Construtores; 2.1.10 Sobrecarga de métodos; 2.1.11 Sobrescrita de métodos; 2.1.12 Agregação e composição. 2.2 Uso do Diagrama de Classes da UML para modelagem de soluções OO. 2.3 Aplicação dos conceitos utilizando uma linguagem de programação orientada a objetos.
3	Classes abstratas e interface	3.1 Métodos virtuais puros; 3.2 Conceito de interfaces.

4	Arrays e coleções	4.1 Aplicação de OO na construção de estruturas de dados.
5	Tratamento de exceções	5.1 Try-catch; 5.2 Desempenho computacional de exceções; 5.3 Casos de uso para exceções.

<b>PROFESSOR</b>	<b>TURMA</b>
Daniel Cavalcanti Jeronymo	CP44

ANO/SEMESTRE	CARGA HORÁRIA (aulas)					
	AT	AP	APS	AD	APCC	Total
2017/1º	33	33	04	00	00	70

AT: Atividades Teóricas, AP: Atividades Práticas, APS: Atividades Práticas Supervisionadas, AD: Atividades a Distância, APCC: Atividades Práticas como Componente Curricular.

DIAS DAS AULAS PRESENCIAIS						
Dia da semana	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
Número de aulas no semestre	34	00	00	00	32	00

PROGRAMAÇÃO E CONTEÚDOS DAS AULAS (PREVISÃO)		
Dia/Mês ou Semana ou Período	1 Conteúdo das Aulas	Número de Aulas
1º (03/03)	Apresentação do plano de ensino.	2
2º (06/03)	Revisão de algoritmos e fundamentos de programação.	2
3º (10/03)	<b>Lista de exercícios.</b>	2
4º (13/03)	Abstração em computação, paradigmas de linguagens de programação. Conceitos de orientação a objetos. Introdução ao C++.	2
5º (17/03)	<b>Lista de exercícios.</b>	2
6º (20/03)	Fundamentos de C++: origens, padrões ISO, diferenças para C, comentários, cabeçalhos, tipos primitivos e compostos, operador de escopo, entrada e saída, definição de variáveis, variáveis referência, classes <i>string</i> e <i>vector</i> , <i>namespaces</i> .	2
7º (24/03)	<b>Lista de exercícios.</b>	2
8º (27/03)	<b>Semana Acadêmica de Engenharia e Tecnologia.</b>	2
9º (31/03)	Classes e instâncias. Construtores, destrutores e operador de atribuição de cópia. Objetos, atributos, operações: mensagens e métodos, estados. Encapsulamento, ocultamento. <b>Lista de exercícios.</b>	2
10º (03/04)	Herança.	2
11º (07/04)	<b>Lista de exercícios.</b>	2
12º (10/04)	<b>Definição do projeto (APS).</b>	2
13º (17/04)	Exemplos de bibliotecas gráficas para a APS.	2
14º (24/04)	Laboratório para dúvidas e exercícios.	2
15º (28/04)	Laboratório para dúvidas e exercícios.	2
16º (05/05)	<b>A<sub>1</sub> – 1ª avaliação.</b>	2
17º (08/05)	Vista e correção da A <sub>1</sub> – 1ª avaliação.	2
18º (12/05)	Exercício de abstração aplicada a OO.	2

<b>PROGRAMAÇÃO E CONTEÚDOS DAS AULAS (PREVISÃO)</b>		
<b>Dia/Mês ou Semana ou Período</b>	<b>1 Conteúdo das Aulas</b>	<b>Número de Aulas</b>
19º (15/05)	Polimorfismo. Polimorfismo estático e dinâmico. Métodos virtuais, interfaces.	2
20º (19/05)	<b>Lista de exercícios.</b>	2
21º (22/05)	Templates, metaprogramação.	2
22º (26/05)	<b>Lista de exercícios.</b>	2
23º (29/05)	Diagramas UML, uso, modelagem direta e inversa.	2
24º (02/06)	<b>Lista de exercícios.</b>	2
25º (05/06)	<b>Laboratório para dúvidas e exercícios.</b>	2
26º (09/06)	<b>A<sub>2</sub> – 2ª avaliação.</b>	2
27º (12/06)	Vista e correção da A <sub>2</sub> – 2ª avaliação.	2
28º (19/06)	Laboratório para desenvolvimento da APS.	2
29º (23/06)	<b>Prazo de entrega do projeto (APS).</b>	2
30º (26/06)	<b>A<sub>E</sub> – Avaliação Extra.</b>	2
31º (30/06)	Vista e correção da A <sub>E</sub> – Avaliação extra.	2
32º (03/07)	Publicação das notas finais.	2
33º (07/07)	Encerramento da disciplina.	2

<b>PROCEDIMENTOS DE ENSINO</b>
<b>AULAS TEÓRICAS</b>
<b>Aulas expositivas e dialogadas, com resolução de exercícios.</b>
Será utilizado como referencial teórico de estudo a bibliografia sugerida, devendo ser enfatizados e trabalhados pelo docente os principais pontos. Ao final de cada assunto, ou conjunto de assuntos relacionados, será proposto aos discentes o desenvolvimento de exercícios individuais ou em dupla para fixação da teoria apresentada.
<b>AULAS PRÁTICAS</b>
<b>Em laboratório de informática, com a implementação computacional dos exercícios.</b>
De forma complementar, serão desenvolvidas implementações e práticas através do uso de ferramentas de <i>software</i> no Laboratório de Informática, traçando uma ligação entre os conceitos generalistas e sua aplicação em situações específicas da futura atuação profissional do discente.
<b>ATIVIDADES PRÁTICAS SUPERVISIONADAS</b>
Será proposto um projeto que contemple a aplicação conjunta dos principais conteúdos trabalhados durante a disciplina, visando a integração das competências do conteúdo programático. O objetivo da APS é instigar o aluno a aprender a produzir o próprio conhecimento e criar soluções originais, indo além da simples reprodução do conhecimento adquirido em sala de aula, objetivando assim a formação de profissionais capacitados a "aprender a aprender". O projeto deverá ser individual ou em grupo, com no máximo três alunos. Além das aulas práticas, o professor estará disponível em seu horário de atendimento para supervisionar a execução de tal projeto. O projeto será avaliado em data a ser definida como atividade para integralizar a nota final.
<b>ATIVIDADES A DISTÂNCIA</b>
Não há.
<b>ATIVIDADES PRÁTICAS COMO COMPONENTE CURRICULAR</b>
Não há.

## **PROCEDIMENTOS DE AVALIAÇÃO**

### A avaliação será composta por: provas, trabalhos e APS.

Para avaliação abrangente do discente serão observados os seguintes pontos:

- a) assiduidade;
- b) participação em aula;
- c) implementação de trabalhos no Laboratório de Informática;
- d) entrega de listas de exercícios resolvidas e
- e) provas escritas abordando o conteúdo trabalhado.

O processo de avaliação ocorrerá por meio de verificações individuais e presenciais, além de tarefas individuais e/ou em grupo a serem desenvolvidas no decorrer das aulas ou como atividades extra-sala, com possíveis apresentações dos trabalhos realizados.

Todas as atividades de avaliação serão pontuadas de 0 (zero) a 10 (dez) pontos, contudo, com pesos diferenciados. A cada atividade será atribuído um peso de acordo com sua relevância e nível de dificuldade, condizente com a complexidade para sua realização, tal que a soma dos pesos totalize 100% (um).

No semestre haverá duas provas,  $A_1$  e  $A_2$ , sendo cada uma individual, escrita e/ou prática e sem consulta a materiais de apoio. O peso de cada uma dessas avaliações na média semestral é 35% (trinta e cinco por cento).

A Atividade Prática Supervisionada (APS) consistirá de um projeto cujo tema e especificações será definido em data estabelecida de acordo com a programação das aulas. O projeto deve ser desenvolvido individualmente ou por equipes de, no máximo, três alunos. O peso desta atividade na média semestral é de 20% (vinte por cento).

A avaliação das listas ( $A_L$ ) será individual e sua pontuação será proporcional a quantidade de listas entregues e devidamente resolvidas. As listas serão resolvidas e pontuadas em sala de aula, em dias específicos para estas atividades. O peso desta avaliação será de 10% (dez por cento) na média semestral.

A média semestral (MS) é dada por:

$$MS = A_1 \cdot 35\% + A_2 \cdot 35\% + APS \cdot 20\% + A_L \cdot 10\% .$$

Os critérios para aprovação (aproveitamento e frequência) seguirão as diretrizes aprovadas e vigentes na instituição para o referido curso.

Para os alunos que obtiverem média semestral (MS) inferior a 6,0 (seis) será possibilitada uma avaliação extra ( $A_E$ ), gerando uma nova média semestral final ( $MS_F$ ) dada por:

$$MS_F = \frac{MS + A_E}{2} .$$

A avaliação extra  $A_E$  abordará todo o conteúdo do semestre.

### REFERÊNCIAS

#### Referências Básicas:

DEITEL, Harvey M.; DEITEL, Paul J. C++ como programar. 5. ed. Porto Alegre, RS: Pearson Prentice Hall, 2006. xlii, 1163 p.

ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. Fundamentos da programação de computadores: algoritmos, Pascal, C/C ++ e Java. 2. ed. São Paulo: Prentice-Hall, c2008. 434 p. ISBN 9788576051480.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo.** 3. ed. Porto Alegre: Bookman, 2007.

#### Referências Complementares:

SILVA FILHO, Antonio Mendes da. Introdução à programação orientada a objetos com C++. Rio de Janeiro: Elsevier, Campus, 2010. 283 p.

STROUSTRUP, Bjarne. A linguagem de programação C++. 3. ed. Porto Alegre: Bookman, 2000. 823 p.

MCLAUGHLIN, Brett; POLLICE, Gary; WEST, David. Use a cabeça análise e projeto orientado ao objeto. Rio de Janeiro, RJ: Alta Books, c2007. 442 p.

KOFFMAN, Elliot B; WOLFGANG, Paul A. T. Objetos, abstração, estruturas de dados e projeto usando C++.

Rio de Janeiro, RJ: LTC, 2008. 689 p.

SUTTER, Herb. Programação avançada em C++: 40 novos quebra-cabeças de engenharia, problemas de programação e soluções . São Paulo: Pearson Education do Brasil, 2006. 289 p.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. 3. ed. Porto Alegre: Bookman, 2007.

#### **ORIENTAÇÕES GERAIS**

A distribuição do conteúdo das aulas é uma previsão, podendo ocorrer ajustes em função do desempenho da turma ou para tratar eventualidades.

Os materiais vistos em sala de aula serão disponibilizados na página pessoal do professor: <http://paginapessoal.utfpr.edu.br/danielc> .

Poderá ser utilizado o ambiente *Moodle* como canal de comunicação entre professor e aluno.

Os horários de atendimento presenciais serão informados previamente pelo professor.

O acesso aos laboratórios em horário pós-aula deverá ser solicitado ao professor e ter anuência do responsável pelo laboratório.

---

Assinatura do Professor

---

Assinatura do Coordenador do Curso