



## Ficha 2 (variável)

<b>Disciplina:</b> Prática em Desenvolvimento de Software/ Estrutura de Linguagens de Programação/Tópicos Especiais I					<b>Código:</b> CI1353/CI214/INFO7070	
<b>Natureza:</b> ( ) Obrigatória ( x ) Optativa		( X ) Semestral ( ) Anual ( ) Modular				
<b>Pré-requisito:</b>		<b>Co-requisito:</b>	<b>Modalidade:</b> ( ) Presencial ( ) Totalmente EaD ( ) ____ *c.H.EaD			
<b>CH Total: 60h</b> <b>CH semanal: 4h</b>		Padrão (PD): 30	Laboratório (LB): 30	Campo (CP): 00	Estágio (ES): 00	Orientada (OR): 00 Prática Específica (PE): 00
		Estágio de Formação Pedagógica(EPP): 00	Extensão (EXT): 00	Prática como Componente Curricular (PCC): 00		
<b>EMENTA (Unidade Didática)</b>						
Histórico. Classes e objetos. Funções e dados membro. Modificadores de acesso. Construtores e destrutores. Alocação dinâmica de memória. Herança e polimorfismo. Templates. Bibliotecas padrão. Lançamento e tratamento de exceções						
<b>1 PROGRAMA (itens de cada unidade didática)</b>						
<b>Aula</b>	<b>CONTEÚDO</b>					
<b>1</b>	- Apresentação do disciplina. Definição das regras, provas, notas, apresentação da bibliografia. - Introdução ao C++.					
<b>2</b>	- Classes e objetos. - Funções e dados membro.					
<b>3</b>	- Modificadores de acesso básicos.					
<b>4</b>	- Construtores.					
<b>5</b>	- Ponteiros, e alocação dinâmica de memória.					
<b>6</b>	- Referências e o ponteiro <i>this</i> .					
<b>7</b>	- A biblioteca STL - Conceitos básicos. - Uso de iteradores.					
<b>8</b>	- Associações.					
<b>9</b>	- Classes e funções amigas.					
<b>10</b>	- Membros Estáticos					
<b>11</b>	- Destrutores					
<b>12</b>	- <b>Prova 1</b>					
<b>13</b>	- Agregações. - Funções e parâmetros const.					
<b>14</b>	- Retornos e ponteiros const. - Funções e dados estáticos. - <b>Definição do trabalho.</b>					
<b>15</b>	- Polimorfismo paramétrico -Templates.					
<b>16</b>	- Herança simples.					
<b>17</b>	- Modificador de acesso <i>protected</i> .					

	- Sobrecarga de funções.
18	- <i>Funções Virtuais.</i>
19	- Herança Múltipla.
20	- Lançamento e tratamento de exceções.
21	- Hierarquia de exceções nas bibliotecas padrão.
22	- Ponteiros Inteligentes.
23	- Sobrecarga de operadores básica.
24	- Sobrecarga de operadores avançada.
25	- Tipos de Casting.
26	- Boost C++.
27	- Prova 2.
28	- Apresentações de trabalhos.
29	- Apresentações de trabalhos.
30	<b>2<sup>a</sup>-chamada Provas 1 e 2</b>
31	<b>Exame Final</b>

#### OBJETIVO GERAL

Apresentar ao aluno os principais conceitos relacionados à orientação a objetos de maneira prática, através de uma linguagem como o C++. O custo computacional envolvido ao se aplicar os conceitos e boas práticas relacionadas a orientação a objetos serão avaliados quando possível. Ao final da disciplina, o aluno deverá ser capaz de implementar programas utilizando orientação a objetos considerando as diferentes classes e objetos de um sistema e suas interações.

#### OBJETIVO ESPECÍFICO

1. O aluno será capaz de entender conceitos relacionados a orientação a objetos, como o conceito de classes, objetos e encapsulamento.
2. O aluno será capaz de criar programas utilizando orientação a objetos.
3. O aluno será capaz de modelar interações entre diferentes objetos de um sistema.
4. O aluno será capaz de utilizar e estender as classes disponíveis nas bibliotecas padrão da linguagem.
5. O aluno entenderá os conceitos básicos relacionados aos custos extras (*overhead*) de se utilizar a orientação a objetos.

#### PROCEDIMENTOS DIDÁTICOS

1. **Aulas:** Aulas expositivo dialogadas presenciais, onde serão apresentados os conteúdos curriculares teóricos e exemplos de aplicação dos conceitos. Serão utilizados quadro de giz e projetor multimídia. Aulas em laboratório de informática servirão para que os alunos implementem os conceitos apresentados.
2. **Sistema de comunicação:** O Moodle será utilizado como meio de comunicação principal extraclasse.
3. **Material didático específico:** Serão utilizados documentos digitalizados como material de referência básico sobre orientação a objetos e linguagem de programação C++. Também serão disponibilizados links para sites existentes para exercitar os conceitos básicos de algoritmos e eventualmente materiais já disponíveis em MOOC's (Cursos Massivos Abertos). O professor



também poderá produzir vídeos próprios onde serão esclarecidos aspectos específicos ou avançados que possam surgir no decorrer da disciplina.  
Para a compilação e execução dos programas serão usados aplicativos com licença de uso livre para computadores, como o *GNU Compiler Collection (GCC)*.

#### FORMAS DE AVALIAÇÃO

Deverão ser feitas 2 (duas) provas e um trabalho prático.

30% - Prova 1;  
40% - Prova 2;  
30% - Trabalho;

Os critérios para aprovação com ou sem exame final seguirão o disposto na Resolução 37/97-CEPE. Capítulo X, Seção I – Normas Gerais de Avaliação.

#### Sugestão

Apesar da disciplina não possuir pré-requisitos, sugere-se que o aluno tenha proficiência em programação em linguagem C (programação estruturada) ou similares, conhecendo conceitos como ponteiros e structs.

#### Alterações na Ficha

Esse documento pode sofrer ajustes no decorrer da disciplina. Quaisquer alterações serão comunicadas aos alunos com antecedência de pelo menos uma semana.

#### BIBLIOGRAFIA BÁSICA

- [1] Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley, 2013.
- [2] Gamma, E. Padrões de Projetos: Soluções Reutilizáveis. Brasil: Bookman. 2009.
- [3] Deitel, H. M., Deitel, P. J. C++: como programar. 5ª ed. Brasil: Pearson Prentice Hall. 2006.
- [4] *ISO/IEC 14882:2020 Programming languages — C++*: URL: <https://www.iso.org/obp/ui/#iso:std:iso-iec:14882:ed-6:v1:en> . Draft público disponível em <https://isocpp.org/files/papers/N4860.pdf>

#### BIBLIOGRAFIA COMPLEMENTAR

- [1] Schildt, H.; Mayer, R. C completo e total. Pearson Education do Brasil, 1996. ISBN 9788534605953.
- [2] Meyers, S. D. C++ Eficaz: 55 Maneiras de Aprimorar Seus Programas e Projetos. Brasil: Bookman. 2011.
- [3] Aguiar, L. J. Programação em C ++ - 2.ed.: Algoritmos, estruturas de dados e objetos: Bookman Editora. 2008.
- [4] Votre, V. P. C++: Explicado e Aplicado. Brasil: Alta Books Editora. 2016.
- [5] C++ Language Tutorial. 2020. URL: <http://www.cplusplus.com/doc/tutorial>

**Professor da Disciplina:** Prof. Dr. Paulo Ricardo Lisboa de Almeida

**Assinatura:** \_\_\_\_\_

**Chefe de Departamento ou Unidade equivalente:** Prof. Dr. Fabiano Silva



Ministério da Educação  
UNIVERSIDADE FEDERAL DO PARANÁ  
Setor de Ciências Exatas  
Departamento de Informática

Assinatura: \_\_\_\_\_



## **CI1353/INFO7052 - Prática em Desenvolvimento de Software/Técnicas em Modelagem de Aplicações**

### **I. Contato do Professor responsável:**

paulo@inf.ufpr.br