

Ficha 2 (variável)

Disciplina: Prática em Desenvolvimento de Software/Tópicos Especiais I (2021-1 ERE)				Código: CI1353/INFO7070		
Natureza: () Obrigatória (x) Optativa		(X) Semestral () Anual () Modular				
Pré-requisito:		Co-requisito:		Modalidade: () Presencial () Totalmente EaD () ____ *c.h.EaD (X) 100% ERE (Ensino Remoto Emergencial), Res. 22/21-CEPE		
CH Total: 60h CH semanal: 5h		Padrão (PD): 60	Laboratório (LB): 00	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		

EMENTA (Unidade Didática)

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. Introdução a *design patterns* (Padrões de Projeto).

PROGRAMA (itens de cada unidade didática)

Semana	Data	CONTEÚDO
1	20/set-26/set	- Apresentação do disciplina. Definição das regras, provas, notas, apresentação da bibliografia. Ambientação dos alunos ao Moodle e recursos a serem usados durante a disciplina. - Introdução ao C++. - Classes e objetos. - Funções e dados membro.
2	27/set-03/out	- Modificadores de acesso básicos. - Construtores.
3	04/out-10/out	- Ponteiros, Referências e alocação dinâmica de memória. - A biblioteca STL - Conceitos básicos.
4	01/out-17/out	- Uso de iteradores. - Associações.
5	17/out	- Classes e funções amigas.
	19/out	- Prova 1
6	25/out-31/out	- Destrutores. - Agregações. - Funções e parâmetros const.
7	01/nov-07/nov	- Retornos e ponteiros const. - Funções e dados estáticos. - Definição do trabalho – Design Patterns.
8	08/nov-14/nov	- Templates. - Herança.
9	14/nov-21/nov	- Modificador de acesso <i>protected</i> . - Sobrecarga de funções.

		- O modificador <i>virtual</i> .
10	22/nov-28/nov	- Herança Múltipla. - Lançamento e tratamento de exceções.
11	29/nov-05/dez	- Hierarquia de exceções nas bibliotecas padrão. - Apresentação dos trabalhos.
12	06/dez	- Prova 2.
	08/dez	- Finalização.
13	14/dez	2ª-chamada Provas 1 e 2
14	21/dez	Exame Final

OBJETIVO GERAL

Apresentar ao aluno os principais conceitos relacionados à orientação a objetos de maneira prática, aplicando-a 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 do sistema e suas interações.

OBJETIVO ESPECÍFICO

1. Além tomar conhecimento do conteúdo e formas de avaliação do curso, o aluno estará preparado para utilizar as ferramentas disponibilizadas para as atividades remotas, e terá entendimento da dinâmica de condução da disciplina.
2. O aluno será capaz de entender o conceitos básicos relacionados a orientação a objetos, como o conceito de classes, objetos e encapsulamento.
3. O aluno será capaz de criar programas utilizando orientação a objetos.
4. O aluno será capaz de modelar interações entre diferentes objetos de um sistema.
5. O aluno será capaz de utilizar e estender as classes disponíveis nas bibliotecas padrão da linguagem.
6. O aluno entenderá os conceitos básicos relacionados aos custos computacionais de se utilizar a orientação a objetos.
7. O aluno será capaz de utilizar conceitos básicos relacionados a padrões de projetos em seus softwares.

PROCEDIMENTOS DIDÁTICOS

1. **Sistema de comunicação:** Na fase inicial da disciplina deverá ser usado e-mail para comunicação com os alunos com vistas a introduzi-los no uso do Moodle da UFPR Virtual e do ambiente da disciplina. Também nesta fase serão realizados levantamentos junto aos alunos quanto ao melhor ambiente de videoconferência para atividades síncronas (Teams ou BigBlueButton). Vencida esta fase, com os alunos já adequadamente acessando a área da disciplina na UFPR Virtual, passamos a usar o mecanismo de Fórum para comunicações com e entre os alunos.
2. **Atividades síncronas:** As atividades síncronas consistirão de aulas ao vivo (*online*), com duração de **1h30m**. Durante as aulas, o conteúdo será apresentado. As atividades síncronas serão gravadas e ficarão disponíveis em local restrito ao aluno, com o objetivo de permitir que os alunos tenham acesso posterior à atividade, caso por algum motivo não lhe tenha sido possível a participação no dia e horário programados.
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)*.

4. **Infraestrutura de suporte tecnológico, científico e instrumental à disciplina:** A UFPR possui um plano para disponibilizar computadores e acesso à Internet aos alunos com problemas de acessibilidade digital.
Os exercícios práticos de programação poderão ser executados em computadores comuns, com a utilização de ambientes de programação com licença de uso livre e disponíveis para os principais sistemas operacionais.
5. **Previsão de período de ambientação dos recursos tecnológicos a serem utilizados pelos discentes:** Haverá na primeira semana de aula a disponibilização de tutoriais e vídeos curtos indicando como deverá ser o andamento da disciplina utilizando a UFPR Virtual. Inicialmente, os alunos serão avisados via mensagem utilizando as ferramentas oficiais da Universidade (e.g., SIGA) sobre como acessar o Moodle da disciplina. Haverá também neste momento uma atividade síncrona de forma a esclarecer dúvidas e ajudar na solução dos eventuais problemas que possam ocorrer com o acesso on-line e instalação de programas.
6. **Identificação do controle de frequência das atividades:** O controle de frequência será feito com base no cumprimento de todas as atividades **avaliativas, dentro do prazo**. O cumprimento das metas estabelecidas nestas atividades contarão como presença, **desde que haja completude e coerência com os conteúdos da disciplina e com o que é solicitado na atividade avaliativa**.

A quantidade total de atividades avaliativas corresponderá à carga horária total da disciplina. A frequência final em horas será calculada proporcionalmente de acordo com a fórmula:

Frequência = (Qtde. atividades avaliativas entregues * CHT) / Qtde. total de atividades avaliativas

7. **Carga horária semanal para atividades síncronas e assíncronas:** As atividades síncronas e assíncronas serão distribuídas da seguinte forma:
 - Atividades síncronas: **3h00m** por semana (em média).
 - Atividades assíncronas: **2h00m** por semana (em média). Essas atividades contemplam, entre outros, a resolução dos exercícios dados em aula e estudos/pesquisas referentes ao conteúdo ministrado.

FORMAS DE AVALIAÇÃO

Deverão ser feitas 2 (duas) provas e a cada semana haverá um conjunto de exercícios que deverão ser entregues pelo aluno dentro de um prazo estipulado, sendo que os exercícios semanais serão assíncronos.

Para a validação das provas e exercícios, os alunos poderão ser chamados, a critério do professor, em um momento síncrono dentro do ambiente Moodle. Serão usados sistemas de detecção de similaridade nas produções dos alunos. Em se verificando similaridades e plágio, os alunos envolvidos serão chamados pelo professor e poderão receber nota 0 (zero), conforme regimentos vigentes na UFPR.

Provas não realizadas pelo aluno são passíveis de 2ª-chamada, nos casos amparados pelo artigo 106, Seção V, Resolução 37/97-CEPE, e considerando também o disposto no artigo 12, § 7º e 8º, Resolução 22/21-CEPE, em data e local divulgados no Cronograma da disciplina.

Em cada semana serão submetidos ao aluno exercícios que deverão ser entregues dentro do prazo estipulado nos respectivos enunciados. A nota final de exercícios (**EXERCIS**) será a média aritmética das notas de todos os exercícios. Exercícios não entregues terão nota 0 (zero).

Os alunos também deverão apresentar um trabalho final, a ser especificado pelo professor.

As provas e exercícios são **atividades avaliativas** e serão contabilizadas na frequência conforme indicado no item 6 dos **Procedimentos Didáticos**. A entrega da atividade será contabilizada ao receber a nota do professor.

Não serão aceitas entregas de exercícios após o final do Período Especial.

A média final será calculada da seguinte forma:



20% - nota obtida pela média aritmética das notas dos exercícios semanais;
20% - nota obtida pelo trabalho final;
30% - nota obtida na Prova 1;
30% - nota obtida na Prova 2.

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.

BIBLIOGRAFIA BÁSICA (mínimo 03 títulos)

- [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>

BIBLIOGRAFIA COMPLEMENTAR (mínimo 05 títulos)

- [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] Aguilar, 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

Assinatura: _____



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

I. Período atividades: de 20/09 a 09/12 [12 semanas]

- 3h00min (em média) aulas síncronas/semana.
- 2h00min (em média) atividades assíncronas/semana.

II. Plano de Ensino e Cronograma da disciplina

Vide **Ficha 2**.

III. Turmas e vagas e professor responsável:

CURSO	TURMA	VAGAS	Horário aulas síncronas
Ciência da Computação	A	15	Segundas e Quartas , 17h30-19h00
Pós Graduação	B	15	

IV. Contato do Professor responsável:

paulo@inf.ufpr.br