



PLANO DE ENSINO

CURSO	<u>Engenharia Eletrônica</u>	MATRIZ	
-------	------------------------------	--------	--

FUNDAMENTAÇÃO LEGAL	(Resolução do COEPP que aprovou a matriz curricular do curso e, se houver, resoluções posteriores relativas à disciplina/unidade curricular) - SA
---------------------	---

DISCIPLINA/UNIDADE CURRICULAR	CÓDIGO	PERÍODO	CARGA HORÁRIA (horas)		
			Teórica	Prática	Total
<u>Fundamentos de Programação 2</u>	IF62C	2º	30	30	60

PRÉ-REQUISITO	<u>Fundamentos de Programação 1 - IF61A</u>
EQUIVALÊNCIA	Relativo – depende de análise pontual de cada ementa.

OBJETIVOS

Instruir o aluno a Programação Orientada a Objetos (POO), a princípios de Programação Orientada a Eventos, a princípios de Programação Orientada a Componentes, a princípios de Interfaces de Programação de Aplicação (*Application Programming Interfaces – APIs*), a princípios de Projeto e ao enquadramento do Projeto/Implementação no Processo de Desenvolvimento de Logiciaria (i.e. *software*). Desenvolver o raciocínio na abordagem de problemas computacionais usando este ferramental. Exercitar os mecanismos deste ferramental utilizando a linguagem de Programação Orientada a Objeto (citando C++ no caso do curso de Engenharia Eletrônica e de Engenharia da Computação) e seus adendos, bem como a sua conexão para com a Linguagem de Unificada de Modelagem (*Unified Modeling Language - UML*).

EMENTA

Estratégias Algorítmicas de Resolução de Problemas. Projeto Orientado a Objetos. Conceitos de Padrões de Projeto e Uso de *APIs*. Ambientes Integrados de Desenvolvimento e de Prototipação. Atividades de laboratório.

CONTEÚDO PROGRAMÁTICO

ITEM	EMENTA	CONTEÚDO
1	Projeto Orientado a Objetos	<ul style="list-style-type: none">- Engenharia de <i>software</i> e suas fases: ênfase nas fases de projeto e implementação.- Conceitos básicos de orientação a objetos: abstração e encapsulamento, classes, objetos, atributos e métodos - em diagramas (de classes) em Linguagem de Unificada de Modelagem (<i>Unified Modeling Language – UML</i>) e em linguagem de Programação Orientada a Objetos (POO)- Gestão do ciclo de vida dos objetos: construtores e destrutores, referência a objetos, relacionamentos entre objetos, persistência- Herança e herança múltipla; Propriedades de classes, atributos e métodos: visibilidade e escopo; Sobrecarga de métodos e operadores e Polimorfismo; Princípio da substituição (ligação dinâmica).- Classes abstratas, Classes aninhadas, Gabaritos (<i>template</i>)

2	Estratégias Algorítmicas de Resolução de Problemas.	<ul style="list-style-type: none"> - Implementação de métodos e/ou relacionamentos de objetos por meio de componentes, tais como listas, filas e pilhas. - Recursividade de métodos. - Programação orientada a eventos; Conceitos de estados e eventos; Comunicação entre objetos via eventos. - Programação concorrente: linhas de Execução (<i>Threads</i>); Problemas de concorrência: sincronização de <i>threads</i>, leituras concorrentes em objetos compartilhados, bloqueios mortais (<i>deadlocks</i>), produtor-consumidor
3	Conceitos de Padrões de Projetos e APIs	<ul style="list-style-type: none"> - Conceito geral de padrões; Conceito de padrões de projeto; Exemplos de padrões de Projeto (e.g. padrões <i>Gang of four - Gof</i>); Aplicações de Padrões de Projeto; - Conceito de Interfaces de Programação de Aplicação ou <i>Application Programming Interface (API)</i> e exemplos: como <i>API</i> de acesso a recursos e serviços do sistema operacional, construção de interfaces gráficas e comunicação em rede
4	Ambientes Integrados de Desenvolvimento e Prototipação	<ul style="list-style-type: none"> - Configuração de Ambiente de Desenvolvimento para Programação Orientada a Objetos (POO). - Programação Orientada a Objetos Gráficos (janelas, botões etc.) e a Eventos; - Ambiente de Programação ‘Visual’ ou ‘Ferramenta Feiticeira’ (<i>Wizard Tool</i>) - Noções de Prototipagem Rápida em Ambientes de Desenvolvimento ou <i>Rapid Application Development (RAD)</i>. - Ferramentas de Engenharia de <i>Software/Sistema</i> Auxiliada por Computador ou <i>Computer Aided Software/System Engineering (C.A.S.E.)</i>. - Integração de Ferramenta <i>C.A.S.E.</i> e Ambientes de Desenvolvimento POO;
5	Atividades de Laboratório	<ul style="list-style-type: none"> - Programação em Ambiente C++ com aplicações <i>console</i>; - Projeto <i>UML</i> (particularmente Diagrama de Classes) em Ferramenta <i>C.A.S.E.</i> conectável ao Ambiente C++; - Programação em Ambiente C++ com suporte a APIs.; - Programação em Ambiente C++ com suporte a objetos gráficos e programação visual, particularmente <i>RAD</i>.

PROFESSOR	TURMA
Prof. Robinson Vida Noronha	S12

ANO/SEMESTRE	CARGA HORÁRIA (aulas)				
	AT	AP	APS	AD	Total
2015/02	32	32	4	X	68

AT: Atividades Teóricas, AP: Atividades Práticas, APS: Atividades Práticas Supervisionadas, AD: Atividades a Distância.

DIAS DAS AULAS PRESENCIAIS

Dia da semana	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
Número de aulas no semestre	28		36			4 - APS

PROGRAMAÇÃO E CONTEÚDOS DAS AULAS (PREVISÃO)

Dia/Mês	Conteúdo das Aulas	N. Aulas
19/08	Apresentação da Disciplina	2
24/08	Introdução à Orientação a Objetos	2
26-08	Criação de Classes	2

PROGRAMAÇÃO E CONTEÚDOS DAS AULAS (PREVISÃO)		
Dia/Mês	Conteúdo das Aulas	N. Aulas
31-08	Herança - Métodos definidos para receber instâncias das classes pai e filhos.	2
02-09	Herança Simples	2
09-09	Exercícios	2
14-09	Semana Tecnológica	2
16-09	Semana Tecnológica	2
21-09	Relacionamentos de Agregação e Composição (definição e implementação na linguagem C++) Autorelacionamento	2
23-09	Classes Abstratas e Métodos Virtuais	2
28-09	Bibliotecas Vector e String, Iterator, Introdução à UML	2
30-09	Diagramas da UML	2
05-10	Realização da Prova como primeira avaliação	2
07/10	Definição do Projeto	2
14/10	Sobrecarga de Métodos e Operadores	2
19-10	Construtores e destrutores	2
21-10	Estudo de Caso: O problema das Oito rainhas	2
26-10	Implicações da Substituição	2
28-10	Herança Multipla	2
04-11	Overload e Overriding	2
09-11	Templates ou Generics	2
11-11	Padrões de Projeto - Observer	2
16-11	Padrões de Projeto - Uso Excessivo de Herança (Decorando Objetos)	2
18-11	Padrões de Projeto - Acoplamento Solto	2
23-11	Padrões de Projeto - Singleton	2
25-11	Padrões de Projeto - Command	2
30-11	Apresentação de Trabalho de conclusão da disciplina.	2
02-12	Apresentação de Trabalho de conclusão da disciplina.	2
07-12	Curiosidades	2
09-12	Curiosidades	2
14-12	Realização do Exame de Recuperação.	2

PROGRAMAÇÃO E CONTEÚDOS DAS AULAS (PREVISÃO)		
Dia/Mês	Conteúdo das Aulas	N. Aulas
16-12	Encerramento da Disciplina	2
APS	<p>Entrada e Saída de Dados em Arquivo Texto: O aluno deverá realizar um estudo de como um conjunto de dados de objetos podem ser lidos e escritos em um documento texto.</p> <p>O resultado desse estudo será uma aplicação que instancie um conjunto de objetos a partir de um documento texto com os dados formatados.</p> <p><i>Exemplo Sugerido:</i></p> <p><i>Formato Sugerido do Arquivo Texto Externo:</i></p> <pre> <pre> <nome> Carlos Henrique </nome> <idade> 34 </idade> </pessoa> <pessoa> <nome> Maria dos Anjos </nome> <idade> 22 </idade> </pessoa> <pessoa> <nome> Julia Maria </nome> <idade> 12 </idade> </pessoa> <pessoa> <nome> Bill Stone </nome> <idade> 47 </idade> </pessoa> </pre>	4

PROCEDIMENTOS DE ENSINO
AULAS TEÓRICAS
O Professor efetivará a aprendizagem ministrando as aulas e os conteúdos corretamente.
O Professor efetivará a aprendizagem ministrando as aulas e os conteúdos corretamente o que inclui o estímulo a atenção, a relação com a aplicação prática, a abertura a questionamentos e questionamentos para turma. Enfim, utilizar-se-ão os bons princípios relativos ao ensino.
AULAS PRÁTICAS
O Professor efetivará a aprendizagem com atividades práticas.
O Professor efetivará a aprendizagem com atividades práticas com a demanda e correção de exercícios e de um trabalho prático que se constituirá na implementação de <i>software</i> de certa complexidade, como um jogo.
ATIVIDADES PRÁTICAS SUPERVISIONADAS
O Professor efetivará a aprendizagem com Atividades Práticas Supervisionadas (APS), definindo exercícios extra para aumentar o aprendizado. As APS serão corrigidas pelo Professor e devolvidas aos respectivos alunos, com o intuito de verificar, instruir e mesmo perceber o andamento dos esforços relativos aos trabalhos.
ATIVIDADES A DISTÂNCIA
Não há aulas à distância.
PROCEDIMENTOS DE AVALIAÇÃO

- **A avaliação será constituída de 2 provas escrita, 1 trabalho prático e APS. Haverá prova substitutiva que poderá substituir a primeira prova do semestre se for maior.**

A nota final será apurada da seguinte forma $NF = (Prova+TF)/2 + APS/10$.

Onde, NF = nota final, Prova = Nota da Primeira Prova ou Nota do Exame de Recuperação, TF = trabalho de fim de disciplina.

Exercício para entregar= Sempre existirá um exercício valendo nota para a APS para ser feito em horário extra e ser apresentado na aula seguinte

REFERÊNCIAS

Referencias Básicas:

- DEITEL, H. M.; DEITEL, P. J. **C++ Como Programar**. 3ª Edição / 5ª Edição. Bookman. 2001.
3ª Edição – 2001: - 10 exemplares (em português) no Campus Curitiba - Central e
- 10 exemplares (em português) no Campus Cornélio Procópio.
5ª Edição – 2006: - 3 exemplares (em português) no Campus Curitiba - Central,
- 7 exemplares (em português) no Campus Campo Mourão,
- 6 exemplares (em português) no Campus Cornélio Procópio e
-12 exemplares (em português) no Campus Guarapuava.
Obs.: Levantamento da quantidade de exemplares feito no início do 2º Semestre de 2012.
- STROUSTRUP, B.
2ª Edição – 1991: - 1 exemplar (em inglês) no Campus Curitiba - Central.
?ª Edição – 1994: - 1 exemplar (em inglês) no Campus Curitiba - Central.
3ª Edição – 1997: - 1 exemplar (em inglês) no Campus Curitiba - Central.
Obs.: Há exemplares (de 1993, 1997 e 2000) em outros *Campi* da UTFPR.
Obs.: Levantamento da quantidade de exemplares feito no início do 2º Semestre de 2012.
- SIMÃO, J. M. **Fundamentos de Programação 2 – C++**. Página de Internet: <http://www.pessoal.utfpr.edu.br/jeansimao/Fundamentos2/Fundamentos2.htm>.
Acesso livre e referenciado inclusive no portal do MEC.
- PRESSMAN, R. S. **Software Engineering – A Practitioner’s Approach**. McGraw Hill (Higher Education). 1987 - 2005.
?ª Edição – 1987: - 01 exemplar (em inglês) no Campus Curitiba - Central.
3ª Edição – 1992: - 02 exemplares (em inglês) no Campus Curitiba - Central.
?ª Edição – 1995: - 21 exemplares (em português) no Campus Curitiba - Central.
6ª Edição – 2005: - 01 exemplar (em inglês) no Campus Curitiba - Central..
6ª Edição – 2006: - 01 exemplar (em português) no Campus Curitiba - Central..
Obs.: Há exemplares (de 1995 e 2002) em outros *Campi* da UTFPR.
Obs.: Levantamento da quantidade de exemplares feito no início do 2º Semestre de 2012.
- GAMMA, E.; HELM, R.; Johnson, R.; Vlissides, J. **Design Patterns: Elements of Reusable Object-oriented Software**. Addison Wesley Longman, 1995 - 2005.
?ª Edição – 1995: - 5 exemplares (em inglês) no Campus Curitiba - Central.
?ª Edição – 2005: - 5 exemplares (em português) no Campus Curitiba - Central.
Obs.: Há exemplares (de 2002 e 2004) em outros *Campi* da UTFPR.
Obs.: Levantamento da quantidade de exemplares feito no início do 2º Semestre de 2012.

Referências Complementares:

- SEBESTA, R. W. **Conceitos de Linguagens de Programação**. 5ª Edição. Bookman 2003.
- HORSTMANN, C. **Conceitos de Computação com o Essencial de C++**, 3ª edição, Bookman, 2003, ISBN 0-471-16437-2.
- RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. **The Unified Modeling Language Reference Manual**. 2nd Edition. Addison-Wesley. 2005. ISBN 0-321-26797-4.
- RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. Addison-Wesley. 2005. ISBN 0-201-57169-2.
- LARGMAN, G. **Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design**. Prentice Hall. 1998. ISBN 0-13-748880-7.
- BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. Editora Campus. 2003. ISBN 85-352-1032-6.

ORIENTAÇÕES GERAIS

Frequência mínima às aulas: 75% do total de aulas ministradas.

Assinatura do Professor

Assinatura do Coordenador do Curso