



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ - ARA

PLANO DE ENSINO

SEMESTRE 2019.2

I. IDENTIFICAÇÃO DA DISCIPLINA:

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS		TOTAL DE HORAS-AULA SEMESTRAIS
		TEÓRICAS	PRÁTICAS	
DEC7532	Linguagem de Programação II	0	4	72

HORÁRIO

TURMAS TEÓRICAS	TURMAS PRÁTICAS	MODALIDADE
	04655 – 6.1420-2 - 6.1620-2	Presencial

II. PROFESSOR(ES) MINISTRANTE(S)

Prof. Antonio Carlos Sobieranski

E-mail: a.sobieranski@ufsc.br

Horário de Atendimento: Segunda-feira das 15-17hrs, Sala C-112, Jardim das Avenidas.

III. PRÉ-REQUISITO(S)

CÓDIGO	NOME DA DISCIPLINA
-	Esta disciplina não possui pré-requisitos

IV. CURSO(S) PARA O(S) QUAL(IS) A DISCIPLINA É OFERECIDA

Graduação em Engenharia de Computação

Bacharelado em Engenharia de Energia

V. JUSTIFICATIVA

O aluno ao final desta disciplina deverá ser capaz de transpor algoritmos, tal como apreendido em lógica de programação, para uma linguagem de programação sob o paradigma da programação orientada por objetos.

VI. EMENTA

Fundamentos de uma linguagem orientada por objetos: nomes, variáveis, tipos de dados, operadores, expressões, estruturas de controle de fluxo, regras de escopo. Decomposição de problemas por objetos. Encapsulamento. Classes: Abstrata, Derivada, Genérica, Aninhada e Agregada. Mecanismo de herança. Polimorfismo. Interfaces. Objetos Componentes. Interface Gráfica com o Usuário (GUI). Sistemas Orientados por Eventos. Mecanismo de resposta à eventos. Persistência: streams, entrada e saída de dados.

VII. OBJETIVOS

Objetivos Gerais: O aluno ao final desta disciplina deverá ser capaz de transpor algoritmos, tal como apreendido em lógica de programação, para uma linguagem de programação sob o paradigma da programação orientada por objetos.

Objetivos Específicos: Motivação para a adoção do paradigma de orientação por objetos. Decomposição de problemas por objetos. Paradigma Estruturado versus Orientado por Objetos. Conceito de Classe. Encapsulamento. Atributos e Comportamento e Modificadores de Acesso. Instância, Objetos e Método Construtor. Ponteiros versus Referências para Objetos. Herança, Interfaces, Classes Abstratas, Classes Aninhadas e Classes Genéricas. Interface com o Usuário em Sistemas Orientados por Eventos. Entrada e Saída de dados em Interfaces com o Usuário. Persistência de Objetos, Streams de Entrada e Saída de Dados para meios persistentes.

VIII. CONTEÚDO PROGRAMÁTICO

UNIDADE 1:

- Apresentação da disciplina. Apresentação do plano de ensino.
- Da Linguagem C para C++, revisão com exemplos práticos
- Motivação para a adoção do paradigma de orientação por objetos.
- Programação Estruturada X Orientada à Objetos
- Variáveis, tipos de dados primitivos, declaração, inicialização, vetores e matrizes.
- Decomposição de problemas por objetos.
- Operadores Relacionais, Operadores Aritméticos, Operadores Lógicos.
- Composição de Operadores de Atribuição.
- Classes, definição, Variáveis Membros (Atributos), Métodos ou Funções (Comportamento), Método Construtor, Modificadores de Acesso, Declaração e Instanciação de Objetos.
- Classes Como Tipo Abstrato de Dados, Encapsulamento e Identidade de Objetos.
- Correlação do Operador new com a Alocação Dinâmica de Memória (Alocação Dinâmica de Tipo Abstratos de Dados).
- Herança, Classes Abstratas, Métodos Abstratos, Hierarquia de Classes, Classe Ancestral e Classe Derivada, Redefinição de Comportamentos Ancestrais.
- Polimorfismo.

UNIDADE 2:

- Interface com o Usuário em Sistemas Orientados por Eventos.
- Classes Genéricas. Interfaces Genéricas. Métodos Genéricos.
- Entrada e Saída de dados em Interfaces com o Usuário (Entrada e Saída de Dados para Console).
- Persistência de Objetos, Streams de Entrada e Saída de Dados para meios persistentes.
- Serialização de Objetos (Interface *Serializable*).
- Atividades práticas: desenvolvimento de trabalho.

IX. METODOLOGIA DE ENSINO / DESENVOLVIMENTO DO PROGRAMA

Atividades práticas no computador, utilizando o ambiente de desenvolvimento.

Aulas expositivas intercaladas com discussões. Desenvolvimento de trabalhos e exercícios.

Apresentação dos trabalhos.

Material de apoio postado no Moodle.

X. METODOLOGIA E INSTRUMENTOS DE AVALIAÇÃO

- A verificação do rendimento escolar compreenderá **frequência e aproveitamento** nos estudos, os quais deverão ser atingidos conjuntamente. Será obrigatória a frequência às atividades correspondentes a cada disciplina, no mínimo a 75% das mesmas (Frequência Suficiente - FS), ficando nela reprovado o aluno que não comparecer a mais de 25% das atividades (Frequência Insuficiente - FI).
- Serão realizadas quatro avaliações, sendo:
 - o **P1**: Prova Teórica e individual, conceitos da Programação Orientada à Objetos
 - o **P2**: Prova Prática individual, implementações Práticas de Programação Orientada à Objetos
 - o **E1**: Trabalhos práticos em C++ a serem solicitados ao longo do semestre, incluindo a etapa de imersão na linguagem C++ (unidade 1) e tutoriais de QT (unidade 2). Pesos intra-grupos a serem definidos de acordo com a complexidade dos trabalhos. Serão na média de 3 entregáveis em C++ e 6 tutoriais em QT.
 - o **E2**: Trabalho prático final de C++ e elaboração de um projeto projeto com interfaces gráficas em QT. Desenvolvimento e Apresentação.

- A Média Final (MF) será calculada da seguinte forma:

$$MF = (P1 + P2 + E1 + E2) / 4$$

- A nota mínima para aprovação na disciplina será $MF \geq 6,0$ (seis) e Frequência Suficiente (FS). (Art. 69 e 72 da Res. nº 17/CUn/1997).
- O aluno com Frequência Suficiente (FS) e média das notas de avaliações do semestre MF entre 3,0 e 5,5 terá direito a uma nova avaliação no final do semestre (REC), exceto as atividades constantes no art.70, § 2º. A Nota Final (NF) será calculada por meio da média aritmética entre a média das notas das avaliações parciais (MF) e a nota obtida na nova avaliação (REC). (Art. 70 e 71 da Res. nº 17/CUn/1997).

$$NF = \frac{(MF + REC)}{2}$$

- Ao aluno que não comparecer às avaliações ou não apresentar trabalhos no prazo estabelecido será atribuída nota 0 (zero). (Art. 70, § 4º da Res. nº 17/CUn/1997)

Observações:

Avaliação de recuperação

- Não há avaliação de recuperação nas disciplinas de **caráter prático** que envolve atividades de laboratório (Res.17/CUn/97).

Nova avaliação

O aluno, que por motivo de força maior e plenamente justificado, deixar de realizar atividades avaliativas previstas no plano de ensino, deverá formalizar pedido à Chefia do Departamento de Ensino ao qual a disciplina pertence, dentro do prazo de 3 (três) dias úteis, apresentando documentação comprobatória. O pedido de nova avaliação deverá ser formalizado na Secretaria Integrada de Departamentos.

XI. CRONOGRAMA PRÁTICO

AULA (semana)	DATA	ASSUNTO
1	05/08 a 10/08	Apresentação da disciplina. De C para C++
2	12/08 a 17/08	Unidade 1
3	19/08 a 24/08	Unidade 1
4	26/08 a 31/08	Unidade 1
5	02/09 a 07/09	Unidade 1
6	09/09 a 14/09	Unidade 1
7	16/09 a 21/09	Unidade 1
8	23/09 a 28/09	Unidade 1
9	30/09 a 05/10	Unidade 1
10	07/10 a 12/10	Avaliação P1 teórica e P2 prática
11	14/10 a 19/10	Unidade 2
12	21/10 a 26/10	Unidade 2
13	28/10 a 02/11	Unidade 2
14	04/11 a 09/11	Unidade 2
15	11/11 a 16/11	Unidade 2
16	18/11 a 23/11	Unidade 2
17	25/11 a 30/11	Unidade 2
18	02/12 a 06/12	Apresentação E2 Interface Gráfica Obs.: não há recuperação para disciplinas essencialmente práticas.

Obs: O calendário está sujeito a pequenos ajustes de acordo com as necessidades das atividades desenvolvidas. Trabalhos práticos E1 serão em média 09 entregáveis e distribuídos ao longo do semestre.

XII. FERIADOS PREVISTOS PARA O SEMESTRE 2019.2:	
DATA	
07/09/2019	Independência do Brasil (Sábado)
12/10/2019	Nossa Senhora Aparecida (Sábado)
28/10/2019	Dia do Servidor Público (Lei nº 8.112 – art. 236) (Sexta)
02/11/2019	Finados (Sábado)
15/11/2019	Proclamação da República (Sexta)
16/11/2019	Dia não letivo (Sábado)
07/09/2019	Independência do Brasil (Sábado)
XIII. BIBLIOGRAFIA BÁSICA	
<ol style="list-style-type: none"> 1. LARMAN, Graig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos e ao desenvolvimento interativo. 3. ed. Porto Alegre: Bookman, 2007. 2. DEITEL, H. M.; DEITEL, P.J. C++ Como Programar. 5ª edição. Pearson, 2006. 3. DEITEL, H. M.; DEITEL, P.J. Java. Como Programar. 6ª edição. Pearson, 2005. 	
XIV. BIBLIOGRAFIA COMPLEMENTAR:	
<ol style="list-style-type: none"> 1. BOOCH, Grady. Object-Oriented Analysis and Design with Applications (3rd Edition), Addison Wesley, 2007. 2. GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. Padrões de Projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000. (18) 3. KRUCHTEN, Philippe. Introdução ao RUP Rational Unified Process. Rio de Janeiro: Ciência Moderna, 2003. 4. MCLAUGHLIN, Brett; POLLICE, Gary; WEST, David. Use a Cabeça! Análise e projeto orientado ao objeto. São Paulo: Alta Books, 2007 5. BORATTI, Isaias Camilo. Programação Orientada a Objetos em Java. Visual Books, 2007. 	
Os livros acima citados constam na Biblioteca Universitária e Setorial de Araranguá. Algumas bibliografias também podem ser encontradas no acervo da disciplina, via sistema Moodle.	
XV. INFRAESTRUTURA E MATERIAIS NECESSÁRIOS:	
<ol style="list-style-type: none"> 1. Laboratório de informática com, no mínimo, um computador por aluno 2. Espaço físico com mesas, cadeiras e tomadas em quantidades adequadas 3. Acesso à internet 4. Datashow que possa ser operado de forma segura, sem risco de acidentes 5. 20 folhas de papel A4 por aluno 6. 10 folhas prova por aluno 7. Quadro branco e canetas 8. Impressão: monocromática e colorida 	
Obs.: A indisponibilidade de infraestrutura/materiais listados pode causar prejuízos ao processo pedagógico, inviabilizando tanto as atividades dos docentes como as dos alunos, podendo, ainda, acarretar em cancelamento de aulas em último caso.	

Digitally signed by Antonio Carlos
Sobieranski:00530580977
Date: 2019.06.25 15:26:45 BRT

Professor da Disciplina

Aprovado na Reunião do Colegiado do Curso em: 16/08/19

Prof. Fabrício de Oliveira Ourique, Ph.D.
Coordenador do Curso de
Eng. de Engenharia de Curso
Portaria 2763/2018/CR

