



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE - CTS
DEPARTAMENTO DE COMPUTAÇÃO – DEC

PLANO DE ENSINO

SEMESTRE 2021.1

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		MODALIDADE
TURMAS TEÓRICAS	TURMAS PRÁTICAS	Remota Assíncrona e Síncrona
02655 – 2.1620-2 / 4.1620-2	02655 – 2.1620-2 / 4.1620-2	
04655 – 3.1620-2 / 5.1620-2	04655 – 3.1620-2 / 5.1620-2	

II. PROFESSOR MINISTRANTE

Prof. Antonio Carlos Sobieranski

E-mail: a.sobieranski@ufsc.br

Horário de atendimento: Segunda-feira e Terça-feira das 14:00 às 15:00 – por vídeo conferência (sala virtual no Moodle)

III. PRÉ-REQUISITO(S)

Linguagem de Programação I

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

Bacharelado em Engenharia de Computação

V. JUSTIFICATIVA

O aluno ao final desta disciplina deverá ser capaz de transpor algoritmos, em consonância com linguagens de Programação I, para uma linguagem de programação que possibilite a implementação do paradigma de orientação à objetos.

VI. EMENTA

Fundamentos do paradigma Orientado à Objetos. Classes e Métodos. Encapsulamento, Herança e Polimorfismo. Modelagem e solução de problemas utilizando os conceitos de orientação a objetos, decomposição por objetos e tipos abstratos de dados. Interface gráfica com usuário (GUI). Análise dos aspectos tecnológicos complementares à solução de problemas (programação orientada a eventos, persistência de dados e objetos, tratamento de exceções).

VII. OBJETIVOS

Objetivo Geral

- Prover ao aluno conhecimentos do paradigma orientado à objetos de modo ao aluno ser capaz de transpor, modelar, implementar algoritmos.

Objetivos Específicos

- Compreender a motivação para a adoção do paradigma de orientação por objetos.
- Conhecer os principais pilares da orientação à objetos, bem como classes, abstração, herança e polimorfismo.

- Decompor problemas segundo o conceito de orientação à objetos
- Implementar o conceito segundo os aspectos da orientação à objetos.

VIII. CONTEÚDO PROGRAMÁTICO

UNIDADE 1:

- Apresentação da disciplina. Apresentação do plano de ensino.
- Da Linguagem C para C++ e Histórico
- Utilização da biblioteca padrão do C++ e diferenças da programação em Linguagem C
- Compilação, flags de compilação e depuração de código. Ferramentas para detecção de *Memoryleak* e *stackoverflow*
- Alocação dinâmica de memória em C++ e inputs/pipe do SO.
- conceito de namespace, `std::string`, `::stringstream`, `::vector`, `::pair`, `::ifstream`, `::ofstream`, and outros necessários da `std`.

UNIDADE 2:

- Motivação para a adoção do paradigma de orientação por objetos.
- Programação Estruturada X Orientada à Objetos
- 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 3:

- 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. COMPETÊNCIAS E HABILIDADES

1. Conceber, especificar, projetar, construir, testar, verificar e validar programas segundo o paradigma orientado à objetos
2. Interpretar e resolver problemas computacionais empregando recursos lógicos e/ou matemáticos

X. METODOLOGIA DE ENSINO / DESENVOLVIMENTO DO PROGRAMA

1. A disciplina será ministrada com aulas expositivas fornecendo os componentes teóricos. Material de apoio postado no Moodle. Desenvolvimento de trabalho e exercícios;
2. Atividades práticas no computador utilizando ferramenta de desenvolvimento em padrão GNU/GCC.

Requisitos de infraestrutura necessários para ministrar as aulas:

- Acesso à Internet;
- Ambiente Virtual de Aprendizagem - Moodle;
- Disponibilidade de um sistema de vídeo conferência.

XI. 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).
- 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 efetuar as avaliações ou não apresentar trabalhos no prazo estabelecido será atribuída nota 0 (zero). (Art. 70, § 4º da Res. nº 17/CUn/1997)

Avaliações

Serão realizadas quatro avaliações, sendo:

- PT:** Prova Teórica e individual, conceitos da Programação Orientada à Objetos. (Atividade assíncrona que se inicia no horário regular da disciplina com prazo máximo para a conclusão de 24 horas).
- PP:** Prova Prática individual, implementações Práticas de Programação Orientada à Objetos. (Atividade assíncrona que se inicia no horário regular da disciplina com prazo máximo para a conclusão de 24 horas).
- T:** 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.
- G:** Trabalho prático final de C++ e elaboração de um projeto projeto com interfaces gráficas em QT. Desenvolvimento e Apresentação. (Atividade assíncrona onde o aluno gravará um vídeo apresentando o trabalho final e realizará a submissão via Moodle)

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

$$MF = PP*0.20 + PT*0.15 + T*0.35 + G*0.30$$

A avaliação no final do semestre (REC) seguirá a mesma regra da avaliação PT.

- O registro de frequência será efetuado para aulas assíncronas e síncronas. Para as aulas síncronas será realizado o registro ao final das aulas. Para as aulas assíncronas, a presença será contabilizada somente com o envio dos trabalhos solicitados a cada aula.

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.

XII. CRONOGRAMA TEÓRICO/PRÁTICO

AULA (semana)	DATA	ASSUNTO
1 ^a	14/06/2021 a 19/06/2021	Un1. Apresentação da disciplina. Introdução ao C++. std::<cin,cout,string,fstream>. Compilação e flags (g++)(-v, -Wall, -O*). Depuração GNU de código e suas flags (gdb)(-g, -ggdb). Exercícios. (Aula síncrona)
2 ^a	21/06/2021 a 26/06/2021	Un1. C++: aspectos da std::<vector, fstream>. Leitura de arquivos. Metodologia de autovalidação dos trabalhos da disciplina. T1: Dictionary (Aula síncrona)
3 ^a	28/06/2021 a 03/07/2021	Un1. C++: aspectos da std::<vector, string, stringstream>. T2: Person (Aula síncrona ou assíncrona)
4 ^a	05/07/2021 a 10/07/2021	Un1. C++: aspectos da std::pair, ::map. Alocação de memória e ponteiros. T3: Histograms (Aula síncrona ou assíncrona)
5 ^a	12/07/2021 a 17/07/2021	Un2. C++ POO: Introdução e fundamentos teóricos. Classes, Objetos, Encapsulamento. T4: Dictionary with classes (hpp e cpp) (Aula síncrona)
6 ^a	19/07/2021 a 24/07/2021	Un2. C++ POO: Classes – Construtores e Destrutores. Sobrecarga e sobrecarga de operadores. T5: Lista genérica em tempo de execução (memory allocation, pointers, *void) (Aula síncrona ou assíncrona)
7 ^a	26/07/2021 a 31/07/2021	Un2. C++ POO: Herança. (Aula síncrona ou assíncrona)
8 ^a	02/08/2021 a 07/08/2021	Un2. C++ POO: Herança. T6: Inheritance (Aula síncrona ou assíncrona)
9 ^a	09/08/2021 a 14/08/2021	Un2. C++ POO: Polymorphism. T7: Polymorphism (Aula síncrona ou assíncrona)
10 ^a	16/08/2021 a 21/08/2021	Avaliação PP e PT (4 ha)
11 ^a	23/08/2021 a 28/08/2021	Un3. C++ GUI: Introdução. C++ com Qt. T8: Tutoriais Level 1 (Aula síncrona)
12 ^a	30/08/2021 a 04/09/2021	Un3. C++ GUI: Introdução. C++ com Qt. T9: Tutoriais Level 2 (Aula assíncrona)
13 ^a	06/09/2021 a 11/09/2021	Un3. C++ GUI. Implementação da Proposta Final GUI (G) mediante aprovação do professor (Aula síncrona)
14 ^a	13/09/2021 a 18/09/2021	Elaboração trabalho final G
15 ^a	20/09/2021 a 25/09/2021	Apresentação G (Aula síncrona)
16 ^a	27/09/2021 a 02/10/2021	Entrega dos Conceitos Finais e REC

Obs: O calendário está sujeito a pequenos ajustes de acordo com as necessidades das atividades desenvolvidas.

XIII. Feriados previstos para o semestre 2021.1:

DATA	
04/09/2021	Data reservada ao Vestibular 2021.2
05/09/2021	Data reservada ao Vestibular 2021.2
06/09/2021	Data reservada ao Vestibular 2021.2
07/09/2021	Independência do Brasil

XIV. BIBLIOGRAFIA BÁSICA

1. CPLUSPLUS Reference Guide. Disponível em <http://www.cplusplus.com/doc/>
2. C++ Object-Oriented Programming [recurso eletrônico] - 1st ed. 1993. / 1993 - (Livros)
3. DEITEL, H. M.; DEITEL, P.J. C++ Como Programar. 5a. edição. Pearson, 2006.

XV. BIBLIOGRAFIA COMPLEMENTAR

1. C++ [recurso eletrônico] : Object-Oriented Data Structures - 1st ed. 1994. / 1994 - (Livros)
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.

Professor da Disciplina

Aprovado na Reunião do Colegiado do Curso em: ____/____/____

Coordenador do Curso