



Universidade Federal de Santa Catarina
Campus Araranguá - ARA
Centro de Ciências, Tecnologias e Saúde
Departamento de Computação
PLANO DE ENSINO

SEMESTRE 2022.1

I. IDENTIFICAÇÃO DA DISCIPLINA

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS - TEÓRICAS	Nº DE HORAS-AULA SEMANAIS - PRÁTICAS
DEC7532	Linguagem de Programação II	0	4
TOTAL DE HORAS - AULA SEMESTRAIS	HORÁRIO TURMAS TEÓRICAS	HORÁRIO TURMAS PRÁTICAS	MODALIDADE
72	02655A - 2.1620-2	02655A - 4.1620-2	Presencial

II. PROFESSOR(ES) MINISTRANTE(ES)

Prof. Antonio Carlos Sobieranski

E-mail: a.sobieranski@ufsc.br

Horário de atendimento: Terça-feira das 15:00 às 16:00

Local: por vídeoconferência ou local físico a ser definido e agendado com o professor.

III. PRÉ-REQUISITO(S)

DEC0012 Linguagem de Programação I

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

ENGENHARIA DE COMPUTAÇÃO [Campus Araranguá]

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++
- 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 Grafica 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.3

IX. COMPETÊNCIAS/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 laboratoriais no computador utilizando ferramenta de desenvolvimento em padrão GNU/GCC.

Requisitos de infraestrutura necessários para ministrar as aulas:

- Datashow/projetor funcionando e com cabos HDMI/SVGA no comprimento adequado;
- Acesso à Internet;
- Laboratório de informática com computadores funcionando e em número adequado a quantidade de alunos;
- Ambiente Virtual de Aprendizagem - Moodle.

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).

- Serão realizadas quatro avaliações, sendo:
 - o **PT**: Prova Teórica e individual, conceitos da Programação Orientada à Objetos
 - o **PP**: Prova Prática individual, implementações Práticas de Programação Orientada à Objetos
 - o **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.
 - o **G**: 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:

$$MF = AP*0.25 + AT*0.20 + T*0.30 + G*0.25$$

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

A nota mínima para aprovação na disciplina será MF >= 6,0 (seis) e Frequência Suficiente (FS) (Art. 69 e 72 da Res. nº 17/Cun/1997).

Complementação de carga horária: a complementação da carga horária da disciplina ocorrerá da seguinte forma: (i) a Semana de Integração Acadêmica será contabilizada como dias letivos, conforme calendário acadêmico de 2022; e (ii) serão solicitados trabalhos de caráter prático-teórico para complementação de carga horária da disciplina.

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

SEMANA	DATAS	ASSUNTO
1	18/04/2022 a 23/04/2022	Un1. Introdução C++. Estrutura programa. <i>Input/Output</i> . Compilação. Depuração.
2	25/04/2022 a 30/04/2022	Un1. Introdução C++. Dados agregados. Sequências.
3	02/05/2022 a 07/05/2022	Un1. Introdução C++. <i>I/O Streams</i> . <i>File Streams</i> . <i>Memory Allocation</i> . Trabalho T1
4	09/05/2022 a 14/05/2022	Un2. C++ POO: Introdução e fundamentos teóricos.
5	16/05/2022 a 21/05/2022	Un2. C++ POO: Classes, Instâncias, Objetos, Encapsulamento.
6	23/05/2022 a 28/05/2022	Un2. C++ POO: Classes – Construtores e Destrutores. Trabalho T2
7	30/05/2022 a 04/06/2022	Un2. C++ POO: Herança.
8	06/06/2022 a 11/06/2022	Un2. C++ POO: Polymorphism.
9	13/06/2022 a 18/06/2022	Un2. Aspectos Gerais POO Trabalho T3
10	20/06/2022 a 25/06/2022	Avaliação AP e AT (4 ha)
11	27/06/2022 a 02/07/2022	Un3. GUI: Introdução. Qt.
12	04/07/2022 a 09/07/2022	Un3. GUI: Introdução. Qt.
13	11/07/2022 a 16/07/2022	Un3. GUI: Introdução. Qt. Trabalho T4

14	18/07/2022 a 23/07/2022	Elaboração trabalho final G
15	25/07/2022 a 30/07/2022	Elaboração trabalho final G
16	01/08/2022 a 03/08/2022	Trabalho Final G REC, Entrega dos Conceitos Finais

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

XIII. FERIADOS PREVISTOS PARA O SEMESTRE

21/04/2022	Tiradentes
04/05/2022	Dia da Padroeira da Cidade (Campus de Araranguá)
16/06/2022	Corpus Christi

XIV. BIBLIOGRAFIA BÁSICA

- LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. Porto Alegre: Bookman, 2007. xiv, 695 p. ISBN 9788560031528.
- DEITEL, Harvey M.; DEITEL, Paul J. C++: como programar. 5. ed. São Paulo: Pearson Prentice Hall, 2006. xlii, 1163 p. ISBN 9788576050568.
- DEITEL, Harvey M.; DEITEL, Paul J. Java como programar. 6. ed. São Paulo: Pearson Prentice Hall, 2005. xl, 1110 p. ISBN 8536301236.

XV. BIBLIOGRAFIA COMPLEMENTAR

- HALTERMAN, Richard L. Fundamentals of C++ Programming. OpenBook. 2019.
- BOOCH, Grady et al. Object-oriented analysis and design with applications. 3rd ed. Upper Saddle River: Addison Wesley, c2007. xxiii, 691 p. ISBN 9780201895513.
- GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. Padrões de projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000. 364p ISBN 8573076100.
- KRUCHTEN, Philippe. Introdução ao RUP: Rational Unified Process. Rio de Janeiro: Ciência Moderna, 2003. xv, 255 p. ISBN 8573932759 (broch.).
- MCLAUGHLIN, Brett; POLLICE, Gary; WEST, David. Use a cabeça: análise e projeto orientado ao objeto. Rio de Janeiro: Alta Books, c2007. 442 p. ISBN 9788576081456.
- BORATTI, Isaias Camilo. Programação orientada a objetos em JAVA. Florianópolis: Visual Books, 2007. 310 p. ISBN 9788575021996.

Obs: 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(a):

Aprovado pelo Colegiado do Curso em ____/____/____

Presidente do Colegiado: