

Professor: Jefferson Leonardo Ayres Brandao	Disciplina / *Unidade Curricular: Programação de soluções computacionais Carga Horária: 160	Curso: Ciência da Computação Período Letivo: 2023/1 <i>* Exclusivo para Unidades Curriculares</i>
-------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------

Objetivo Geral / *Tópico Gerador:
Quais os diferentes tipos de dados que podem ser utilizados no contexto de programação de computadores Qual a diferença entre os paradigmas de programação vigentes Solução computacional com lógica de programação envolvendo entrada processamento e saída de dados Soluções computacionais através do paradigma orientado a objetos Sistemas gerenciadores de banco de dados como parte da solução computacional Como a interação com o usuário e apresentação de dados através de interface gráfica constitui uma solução computacional Utilização de estruturas e tipos complexos de dados no apoio de soluções computacionais Utilização de métodos eficientes para ordenar e recuperar dados

Objetivos Especificos / *Metas de compreensão:
Identificar problemas que tenham solução algorítmica e usar raciocínio lógico para formular soluções utilizando diferentes formas de representação pseudocódigo fluxograma e linguagem de programação Relacionar a orientação a objetos a um paradigma de programação e implementar soluções neste paradigma a partir de ambientes de desenvolvimento apropriados propondo classes contendo métodos e atributos adequados aplicando o conceito de encapsulamento por meio dos modificadores private protected e métodos de acesso conceitos de método construtor de polimorfismo sobrecarga overload e sobreposição override de métodos Aplicar instruções de consulta para armazenamento e recuperação de dados utilizando conexões com SGDBs e linguagem de consulta estruturada Entender como usar as estruturas de dados como pilhas filas e listas de forma adequada para o desenvolvimento de software Utilizar controle de fluxo de execução em algoritmos e programas computacionais para representar regras de negócio e requisitos Compreender e utilizar adequadamente os tipos primitivos e os tipos complexos de dados objetos enum struct de acordo com as especificidades dos problemas computacionais propostos Desenvolver sistemas com interface gráfica para interação do usuário e persistência de dados para sua posterior manipulação Entender os diversos tipos de relacionamento entre objetos Associação Agregação e Herança Entender o conceito de tratamento de exceções Aplicar o tratamento de exceções em fluxo de negócio de forma computacional Implementar a codificação dos diversos tipos de relacionamento entre objetos Associação Agregação e Herança Selecionar algoritmos apropriados de ordenação e busca para situações particulares visando eficiência

Plano Aula / Atividade Discente / *Desempenho de Compreensão:

Avaliação Contínua:
Atividade Avaliativa 1 (A1) 30pts
Avaliação Integrada (AIT) 30pts (Extra)
Atividade Avaliativa 3 (A3) 40pts
Atividade Avaliativa 2 (A2) 30pts

Exigência mínima para aprovação: 70 pontos e 75% de frequência nas disciplinas presenciais.
Tipo de Avaliação: Avaliação e Frequência

Ementa:

Noções de geometria analítica e teoria de conjuntos Álgebra e Lógica booleana Algoritmos e suas representações Lógica de programação entrada, processamento, saída Constantes e variáveis Estruturas de controle: seleção e repetição Funções e procedimentos Passagem de parâmetros por valor e referência Operações com vetores e matrizes Tipos abstratos de dados: pilhas, filas e listas Algoritmos de pesquisa e de ordenação Paradigmas de Linguagens de Programação Programação orientada a objetos: classes, objetos, métodos, atributos, construtores e métodos de acesso e modificadores Encapsulamento, herança, abstração e polimorfismo Tratamento de exceções Manipulação de arquivos e conexão com SGBD (Sistemas Gerenciadores de Banco de Dados) Bibliotecas de interfaces gráficas Linguagem SQL de banco de dados

Metodologia (Não se aplica a Unidades Curriculares, exceto para o curso de Medicina):

Programa:

Algoritmos e suas representações.
Operações matemáticas básicas com vetores e matrizes.
Lógica booleana.
Álgebra linear e equações algébricas.
Sistema numérico binário, decimal, hexadecimal.
Lógica simples de programação (entrada - processamento - saída), constantes e variáveis.
Estruturas de controle: seleção e repetição.
Vetores, modularização e passagem de parâmetros.
Algoritmos de pesquisa e de ordenação.
Tipos abstratos de dados: pilhas, filas e listas.
Programação orientada a objetos: classes, objetos, métodos, atributos, construtores e métodos de acesso e modificadores.
Encapsulamento, herança, abstração e polimorfismo.
Interfaces gráficas e tratamento de exceções.
Implementação de transações de banco de dados (CRUD) utilizando bibliotecas de conexão com SGBD (Sistemas Gerenciadores de Banco de Dados).
Linguagem SQL de banco de dados.

***Certificação:**

Qualificação Profissional em Programador de Soluções Computacionais

***Competências:**

Identificar planejar e desenvolver soluções computacionais para atender às demandas das organizações Organizar e manter os dados manipulados por uma solução computacional em banco de dados a partir de Sistemas Gerenciadores de Banco de Dados Identificar problemas que tenham solução algorítmica Resolver problemas usando ambientes de programação Identificar e analisar requisitos e especificações para problemas específicos e planejar estratégias para suas soluções Especificar projetar implementar manter e avaliar sistemas de computação empregando teorias práticas e ferramentas adequadas Conceber soluções computacionais a partir de decisões visando o equilíbrio de todos os fatores envolvidos

Emitido por: DBADM

Rio de Janeiro, 02/06/2025

Bibliografia Básica:

MARKENZON. Estruturas de dados e seus algoritmos. Rio de Janeiro: LTC, 2010. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/978-85-216-2995-5/>. Acesso em: 17 fev. 2022.

RIBEIRO, João A. Introdução à programação e aos algoritmos. Rio de Janeiro: LTC, 2019. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788521636410/>. Acesso em: 17 fev. 2022.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. Algoritmos: lógica para desenvolvimento de programação de computadores. 25. ed. rev. São Paulo: Érica, 2011. 320 p. ISBN 9788536502212.

Bibliografia Complementar:

HORSTMANN, Cay S.; CORNELL, Gary. Core Java 1. Rio de Janeiro: Alta Books, 2005. E-book. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/1238>

DEITEL, Harvey M.; DEITEL, Paul J. Java: como programar. 6. ed. São Paulo: Pearson Prentice Hall, 2005. xl, 1110 p. + 1 CD-ROM ISBN 8576050196.

CORMEN, Thomas H. Algoritmos: teoria e prática. Rio de Janeiro: Campus, 2002. xvii, 916 p. ISBN 8535209263.

BOND, Martin (Et al). Aprenda J2EE: com EJB, JSP, Servlets, JNDI, JDBC e XML. São Paulo: Editora Pearson, 2003. 962 p. + 1 CD-ROM ISBN 8534614881.

SOUZA, Marco A. Furlan D.; GOMES, Marcelo M.; SOARES, Marcio V.; CONCILIO, Ricardo. Algoritmos e lógica de programação. São Paulo: Cengage Learning Brasil, 2019. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788522128150/>. Acesso em: 17 fev. 2022.