

MCTA028-15 Programação Estruturada turmas A3 e A4 Matutino – 2024.3

Informações Gerais

Horário

Segunda-feira 8:00 - 10:00 semanal laboratório 404-2 bloco A (turma A3).

Segunda-feira 10:00 - 12:00 semanal laboratório 404-2 bloco A (turma A4).

Quarta-feira 10:00 - 12:00 semanal sala 114 bloco A (ambas turmas).

Professor: Carlos da Silva dos Santos

email carlos.ssantos@ufabc.edu.br

sala 503-2 (torre 2, bloco A).

Horário de atendimento: Quarta-feira, 14h00 até 15h00. Caso você não consiga comparecer nesse dia/horário, escreva para o professor sugerindo até 3 horários alternativos.

Objetivos de Aprendizado

Ao final da disciplina, espera-se que os discentes sejam capazes de:

- Criar programas em uma linguagem compilada a partir de uma especificação do problema a ser resolvido.
- Declarar e utilizar funções.
- Desenvolver funções recursivas.
- Declarar novos tipos de dados e utilizá-los em programas.
- Ler e escrever dados em arquivos.
- Compreender a organização de memória na alocação estática e dinâmica.
- Gerenciar memória manualmente, utilizando ponteiros.
- Compreender e empregar estruturas de dados como matrizes e listas ligadas.
- Compreender e implementar algoritmos de ordenação como *selection sort* e *insertion sort*.
- Compreender e empregar técnicas de busca, como busca linear e busca binária.
- Utilizar boas práticas de desenvolvimento de software, quanto à organização do código e documentação.
- Utilizar ferramentas de depuração de software e empregar estratégias efetivas para a identificação de defeitos.

Conteúdo programático

- Introdução a uma linguagem compilada com suporte a alocação de memória.

- Revisão de expressões, comandos condicionais e de repetição, entrada/saída, funções, vetores e matrizes.
- Noções básicas sobre compilação e organização de computadores.
- Depuração e organização de programas.
- Recursão.
- Registros.
- Ponteiros, alocação estática e dinâmica de memória.
- Listas ligadas simples.
- Arquivos texto e binários.
- Algoritmos elementares de ordenação: Bubble Sort, Selection Sort, Insertion Sort.
- Busca linear e busca binária.

Recomendações

- Processamento da Informação.

Bibliografia Básica

- Notas de aula.
- PINHEIRO, F. A. C. Elementos de programação em C. Porto Alegre, RS: Bookman, 2012.
- FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de programação: a construção de algoritmos e estruturas de dados. 3a edição. São Paulo, SP: Prentice Hall, 2005.
- GUSTEDT, J. Modern C. 1a edição. Editora Manning, 2019. Cópia livre disponível em: <https://inria.hal.science/hal-02383654>

Bibliografia Complementar

- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Algoritmos: teoria e prática. 2a edição. Rio de Janeiro, RJ: Campus, 2002.
- KNUTH D. E. The art of computer programming. Upper Saddle River, USA: Addison-Wesley, 2005.

Dinâmica da disciplina

A disciplina é composta pelos seguintes elementos: aulas expositivas, aulas em laboratório, atividades de programação, questionários e outras atividades eventuais. As atividades da disciplina terão o suporte do ambiente virtual de aprendizagem Moodle (<https://moodle.ufabc.edu.br/>).

Com frequência aproximadamente semanal, haverá um questionário sobre o conteúdo estudado. O propósito do questionário é permitir que o aluno avalie seu próprio progresso e também revelar eventuais dificuldades de aprendizado. As aulas de laboratório serão compostas por exposições rápidas sobre conceitos, seguidas por testes, discussões ou atividades de programação. O propósito das aulas de laboratório é promover a interação docente-alunos e entre os próprios alunos para identificar dúvidas e dificuldades de aprendizado.

Avaliação

Durante o quadrimestre, os discentes deverão entregar atividades (questionários e exercícios de programação) por meio da plataforma Moodle. Cada atividade será associada a um conjunto de competências, baseadas nos objetivos de aprendizado da disciplina. As atividades serão avaliadas pelo docente, que devolverá aos alunos comentários e sugestões de aprimoramento. As atividades serão avaliadas com uma escala de dois conceitos: *Aceito* e *Não Aceito*. Algumas das atividades (como questionários) poderão ser refeitas pelos alunos até obter o desempenho suficiente para aceitação.

Discentes que tiverem frequência inferior a 75% das aulas receberão conceito O (reprovação por frequência). Os demais alunos receberão conceitos de acordo com as competências demonstradas em atividades que tenham sido aceitas. A Tabela abaixo serve de guia para associar conceitos e competências:

Conceito	Competências
F	Discente não demonstrou todas as competências especificadas para o conceito D.
D	Declarar tipos e variáveis. Utilizar estruturas de controle condicionais. Utilizar estruturas de repetição. Declarar e utilizar funções. Ler e escrever dados em arquivos. Utilizar vetores em programas.
C	Utilizar matrizes em programas. Implementar e utilizar busca linear. Declarar e utilizar novos tipos de dados. Desenvolver funções recursivas. Gerenciar memória manualmente utilizando ponteiros.
B	Discente demonstrou todas as competências especificadas para o conceito C e pelo menos metade das competências especificadas para o conceito A

Conceito	Competências
A	Implementar e utilizar listas ligadas simples. Implementar e utilizar algoritmos de ordenação: Bubble Sort, Insertion Sort, Selection Sort. Implementar e utilizar busca binária. Organizar código em módulos (bibliotecas). Compilar código que envolva múltiplas bibliotecas. Implementar testes de software. Automatizar os processos de compilação, teste e geração de documentação. Revisar código.

A relação entre conceitos e competências é cumulativa. Isto é, para obter o conceito C é preciso demonstrar todas as competências especificadas para o conceito D e também as especificadas para o conceito C, e assim por diante. Discentes que recebam conceito final D ou F terão direito a uma avaliação de recuperação.

Política de Colaboração e Plágio

Em seu artigo 25, o código de ética da UFABC estabelece que:

“Quanto aos trabalhos acadêmicos, é eticamente inaceitável que os discentes:

I - fraudem avaliações; II - fabriquem ou falsifiquem dados; III - plageiem ou não creditem devidamente autoria; IV - aceitem autoria de material acadêmico sem participação na produção; “

Durante a disciplina, você pode conversar com seus colegas ou terceiros sobre estratégias gerais de resolução de problemas, sintaxe da linguagem C, dificuldades técnicas e outros assunto relacionados à disciplina. A colaboração entre alunos nesses termos é desejável e favorece o aprendizado.

É expressamente proibido:

- Apresentar o trabalho de outra pessoa como se fosse seu, sem atribuir o devido crédito.
- Copiar inteiramente, ou de maneira parcial, trechos não triviais de código de outras fontes (colegas, livros, *sites* da internet, etc), incorporando este código ao seu trabalho. O que constitui código “não trivial” é dependente do contexto, mas, de maneira geral, qualquer código que exceda algumas linhas ou seja crucial para a resolução do problema proposto será considerado “não trivial”.
- Compartilhar soluções de atividades com colegas ou publicá-las de maneira acessível (p.ex. em *sites* públicos, redes sociais, pastas de nuvem compartilhadas).

Caso você tenha colaborado com alguém, recebido alguma sugestão significativa ou tenha incorporado algum material de terceiro em alguma atividade, você deve

creditar nominalmente a fonte/colaborador e documentar a extensão do material incorporado. Este reconhecimento da fonte é um gesto para afastar a suspeição de fraude (passar o trabalho de outra pessoa como se fosse seu) mas não é um salvo conduto para simplesmente copiar o trabalho de outra pessoa.

Em caso de violação das orientações acima, **todos** os envolvidos estarão sujeitos a:

- Invalidação de **todas** as atividades e atribuição imediata do conceito **F**.
- Denúncia à Comissão de Transgressões Disciplinares Discentes da Graduação, que poderá decidir pelas punições cabíveis.

Calendário PE (previsto, sujeito a alterações)

Semana 1

Aulas:

- 30/09 segunda-feira: Introdução ao Ambiente Computacional.
- 02/10 quarta-feira: Apresentação da disciplina. Introdução à linguagem C. Revisão de tipos de dados, expressões e estruturas de controle.

Semana 2

Aulas:

- 07/10 segunda-feira: **Não haverá aula:** interdição do câmpus para eleição municipal.
- 09/10 quarta-feira: Estruturas de Repetição. Funções. Recursão. Exercícios.

Semana 3

Aulas:

- 14/10 segunda-feira: Exercícios sobre estruturas de controle, funções, recursão.
- 16/10 quarta-feira: Vetores e Matrizes. Exercícios.

Semana 4

Aulas:

- 21/10 segunda-feira: Exercícios.
- 23/10 quarta-feira: Tipos definidos pelo programador. Arquivos.

Semana 5

Aulas:

- 28/10 segunda-feira: **Feriado**: dia do funcionário público.
- 30/10 quarta-feira: Organização de Memória. Ponteiros.

Semana 6

Aulas:

- 04/11 segunda-feira: Exercícios: tipos definidos pelo programador e ponteiros.
- 06/11 quarta-feira: Ponteiros.

Semana 7

Aulas:

- 11/11 segunda-feira: Exercícios: ponteiros.
- 13/11 quarta-feira: Alocação dinâmica de memória.

Semana 8

Aulas:

- 18/11 segunda-feira: Exercícios: ponteiros e alocação dinâmica de memória.
- 20/11 quarta-feira: **Feriado**: dia da consciência negra.

Semana 9

Aulas:

- 25/11 segunda-feira: Exercícios de revisão.
- 27/11 quarta-feira: **Congresso da UFABC**: aula a confirmar.

Semana 10

Aulas:

- 02/12 segunda-feira: Exercícios de revisão.
- 05/12 quarta-feira: Ordenação, lista ligada.

Semana 11

Aulas:

- 09/12 segunda-feira: Exercícios: ordenação e listas.
- 11/12 quarta-feira: Busca binária.

Semana 12

Aulas:

- 16/12 segunda-feira: Exercícios de revisão.
- 18/12 quarta-feira: Revisão.

Reposição

As datas de reposição serão usadas para revisão/refazer atividades/recuperação, conforme necessário.

Atenção para datas e horários

- 24/01/2025 sexta-feira: Reposição de Laboratório.
- 28/01/2025 terça-feira: Reposição de Laboratório.
- 31/01/2025 sexta-feira: Reposição de Teoria.