

Sobre a Disciplina Programação Imperativa

Prof. Alberto Costa Neto
DComp/UFS

Sobre a Disciplina PI

- **Disciplina:** Programação Imperativa (COMP0197)
- **Não equivalentes:**
 - Introdução à Ciência da Computação
 - Programação Imperativa (COMP0334)
- **Carga horária:** 90 horas
- **Créditos:** 6

Ementa

- Noções fundamentais: algoritmos, notação e programas. Identificadores, constantes, variáveis e atribuição.
- Tipos primitivos de dados e tipos derivados.
- Operadores, funções e expressões.
- Instruções condicionais, incondicionais e de repetição.
- Estilo de programação estruturada de programas.
- Representação de dados na forma de vetores, matrizes, registros e conjuntos.

Ementa (continuação)

- Procedimentos, funções e passagem de parâmetros.
- Ordenação por seleção e método da bolha.
- Recursividade. Noções de arquivos em programação.
- Ponteiros.
- Uma linguagem imperativa (por exemplo, Pascal ou C).
- Algoritmos numéricos e não numéricos para a solução de problemas de baixa complexidade.
- Aplicações.
- Modularização de programas. Padrões de codificação.

Objetivos

Geral

- Apresentar os conceitos básicos e principais técnicas de desenvolvimento de programas de computador, tornando-o apto a compreendê-los e aplicá-los.

Específicos

- Tornar o aluno capaz de implementar programas básicos usando uma linguagem de programação imperativa.
- Habilitar o aluno a criar programas para executar computação científica na sua área de conhecimento.
- Colocar em prática os conhecimentos aprendidos no curso, desenvolvendo aplicações de pequeno porte em Python.

Conteúdo Programático

1º Unidade

- Motivação para Programar
- Hardware, software e princípios
- Visão Geral da Linguagem Python
- Preparação do Ambiente de Desenvolvimento
- Instruções primitivas: atribuição, entrada e saída
- Expressões
- Tipos
- Comandos Condicionais (if)
- Tratamento de exceções (try / except)
- Funções
- Laço While
- Strings
- Laços For

2º Unidade

- Listas
- Recursividade
- Dicionários
- Tuplas
- Arquivos

Afinal, por que o nome PI?

Vem da denominação do Paradigma que vamos estudar: Paradigma Imperativo

- Você escreve explicitamente as ordens e o computador obedece
- Mais próximo do funcionamento real do computador
- Existem outros paradigmas, como por exemplo:
 - Funcional
 - Orientado a Objetos

Método de Ensino

Metodologia - Presencial

- Conteúdo teórico explicado em sala de aula estará disponível também pela Youtube
- Sistema que permite programar e tem autoavaliação
- Tempo de aula será focado em exercícios e tirar dúvidas

Recursos didáticos e AVA's

Recursos Didáticos

As aulas serão ministradas presencialmente utilizando as seguintes ferramentas:

- **Youtube**, para exposição das videoaulas.
- **Ferramentas de Videoconferência**: Google Meet.
- **Editores de programas**: Repl.it, Notepad++ ou Sublime Text.
- **Interpretador da linguagem Python**, que permite a verificação de erros de sintaxe e execução de programas em Python.
- **Apps** que permitem elaborar, executar e testar programas em smartphones e tablets.
- **Ambientes Virtuais de Aprendizagem (AVA)** SIGAA e Google Classroom
- Questionários com **Problemas de Programação** no site <http://thehuxley.com>

Correção de Questões

- Imagine se seu professor terá como corrigir 100 questões de cada um dos 50 alunos... Façamos as contas: São 5.000 questões!
Supondo que o professor gaste 6 min por questão, seriam necessários 30.000 minutos, ou seja, 500 horas!
- Seria interessante ter uma ferramenta que ajudasse o professor, concordam?



- Uma ferramenta Web que oferece um **banco de problemas de programação** (juiz *on-line*).
- Os **alunos podem enviar soluções** (programas em várias linguagens de programação).
- O **The Huxley executa a solução** com entradas presentes em casos de teste e compara com o resultado esperado.
- Com esta ferramenta o aluno tem um **feedback imediato**

The Huxley



Avaliação

Critério de Avaliação

Através de testes presenciais, obedecendo à fórmula:

$$\text{Nota Final} = (\text{NT1} + \text{NT2}) / 2$$

Onde:

NT1 = Nota do 1º Teste

NT2 = Nota do 2º Teste

Observação: Haverá um teste de reposição no final do semestre apenas para os alunos com falta justificada em algum teste, desde que a justificativa esteja prevista nas normas acadêmicas.

Calendário de Provas

As provas serão realizadas **presencialmente**:

- No horário da aula
- Segundo **calendário** e **orientações** divulgados nos AVA's

Controle de Frequência

Controle de Frequência (Turmas Presenciais)

O aluno é obrigado a estar presencialmente nas aulas.

Assim, a frequência dos alunos será computada através da **Lista de presença**.

Bibliografia

Referências Bibliográficas (Básica)

- **Fundamentos da Programação de Computadores.** Ana Fernanda Gomes Ascencio / Edilene Aparecida Veneruchi De Campos. 3º edição; 2012, Pearson; ISBN 978-8564574168
- **Algoritmos e Lógica de Programação.** Marco A. Furlan de Souza, Marcelo M. Gomes, Marcio V. Soares, Ricardo Concilio. Editora Cengage Learning, 2ª edição, 2011.
- **Algoritmos: Lógica para Desenvolvimento de Programação de Computadores.** José Augusto N. G. Manzano, Jayr Figueiredo de Oliveira. Editora Érica, 17ª edição, 2005.
- **Python Para Todos: Explorando Dados com Python 3.** Charles R. Severance. Publicação independentes; 1º edição, 2020; ISBN: 979-8635191408

Referências Bibliográficas (Complementares)

- Como pensar como um Cientista da Computação usando Python (traduzido). Allen Downey, Jeffrey Elkner, and Chris Meyers. 2002.
- Introdução à Programação com Python. Nilo Ney Coutinho, 2ª edição, 2014, ISBN: 978-85-7522-408-3.
- Python para Desenvolvedores. Luiz Eduardo Borges. Rio de Janeiro; 2010
- Learning to Program Using Python. Cody Jackson. CreateSpace Independent Publishing Platform

Contatos dos Professores

- Alberto Costa Neto – T02 e T03
 - alberto@dcomp.ufs.br
 - albertocn@academico.ufs.br

Como proceder em caso de dificuldade?

- Sempre que identificar alguma dificuldade, dúvida sobre conceitos das videoaulas ou problemas, **entre em contato com o(s) professor(es)** responsáveis pela sua turma.
- Caso não consiga **acessar os AVAs ou sites**, também entre em contato com o(s) professor(es).

Não deixe de tirar suas dúvidas!

E sejam bem-vindos ao curso de PI!!!