



PLANO DE CURSO

IDENTIFICAÇÃO

DISCIPLINA: Programação Imperativa

CÓDIGO: COMP0197

PERÍODO: 2019.1

C.H.: 90

Nº DE CRÉDITOS: 6

C.H. PRÁTICA: 30

PRÉ-REQUISITO(S): -

TURMAS: T04

HORÁRIO: 246M12

PROFESSORES: Alberto Costa Neto / Kalil Araújo Bispo

EMENTA

Noções fundamentais sobre algoritmos e sobre a execução de programas. Análise e síntese de problemas. Identificadores, tipos, constantes, variáveis, tipos. Operadores e expressões. Comandos condicionais e de repetição. Variáveis compostas homogêneas e heterogêneas. Procedimentos, funções e passagem de parâmetros. Noções sobre o uso de arquivos em programação. Algoritmos básicos de ordenação. Recursividade. Uma linguagem imperativa. Convenções de código. Boas práticas de programação.

OBJETIVOS

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

2. 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 PROGRAMADO (2H/AULA)

AULA 01:

- Introdução à disciplina (e-mail e site da disciplina, bibliografia, avaliação).
- O que é programar
- Motivação para programar

AULA 02: Hardware, software e princípios

AULAS 03 e 04: Visão geral da linguagem Python

AULA 05: Preparação do ambiente de desenvolvimento

AULAS 06 e 07:

- Constantes, variáveis, operador de atribuição e palavras reservadas
- Operador de atribuição
- Atribuição múltipla

AULAS 08 e 09:

- Comandos de entrada e saída



- Comentários
- Tipos de dados, Conversão de tipos e operações com String

AULA 10: Expressões numéricas

AULA 11:

- Expressões relacionais e booleanas
- Ordem de avaliação e precedência de operadores

AULAS 12 e 13:

- Comandos condicionais (if)
- Blocos de código (indentação)

AULA 14:

- Comandos condicionais aninhados e múltiplos (if / else / elif)
- Tratamento de exceções (try / except)

AULAS 15 e 16:

- Comando de repetição while
- Comandos break e continue

AULA 17:

- Uso do laço While
- Funções predefinidas e Bibliotecas

AULAS 18 e 19: Exercícios

AULA 20: 1º Teste

AULAS 21 e 22:

- Strings
- Comando de repetição for

AULA 23:

- Uso do Laço For
- Operadores e funções para Strings

AULAS 24 e 25: Definição de funções, retorno de valores, argumentos e parâmetros

AULA 26: Funções recursivas

AULAS 27 e 28:

- Listas
- Funções e Métodos de Listas

AULA 29:

- Aplicações de Listas
- Compreensão de Listas

AULAS 30 e 31: Dicionários

AULA 32: Aplicações de Dicionários

AULAS 33 a 35: Exercícios



AULA 36: 2º Teste

AULA 37 e 38: Tuplas

AULA 39: Aplicações de Tuplas

AULAS 40 a 42: Leitura e gravação de arquivos texto

AULAS 43 a 47: Exercícios

AULA 48: 3º Teste

METODOLOGIA

O conteúdo teórico estará disponível na Internet para que o aluno assista, possa rever e até adiantar o assunto, conforme sua disponibilidade. Além disso, o aluno terá acesso via Internet a um sistema que permite programar e, através da autoavaliação provida por ele, medir seu aprendizado. Com isso, o tempo de aula será focado em exercícios orientados pelo professor e monitor/tutor. Problemas mais complexos serão abordados no final da disciplina, visando preparar o aluno para implementar soluções no computador para problemas que encontrará na atividade profissional. O aprendizado dos alunos será avaliado através de testes.

FORMA DE AVALIAÇÃO

A avaliação será feita através de testes, obedecendo à fórmula: **Nota Final** = $(NT1 + NT2 + NT3) / 3$.

Onde: NT1 = Nota do 1º Teste, NT2 = Nota do 2º Teste e NT3 = Nota do 3º Teste

Observação: NT3 também servirá como reposição de notas para os alunos com falta justificada, desde que esteja prevista nas normas acadêmicas. Caso o aluno tenha feito todos os testes e obtido alguma nota inferior a 5,0, a NT3 a substituirá.

RECURSOS DIDÁTICOS

As aulas serão ministradas em sala de aula e/ou laboratório (caso haja disponibilidade) com auxílio de data show, quadro e as ferramentas para programação de computadores, são elas:

- Editores de programas: 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 Moodle
- Ambiente Virtual de Aprendizagem Móvel Moodley
- Questionários e Atividades via SIGAA
- Questionários com problemas de programação no site <http://thehuxley.com>

BIBLIOGRAFIA

1. Básica:

- Python for Everybody: Exploring Data Using Python 3. Charles R. Severance. CreateSpace Independent Publishing Platform; 1st. ed., 2016; ISBN: 978-1530051120
- Python for Informatics: Exploring Information. Charles R. Severance. CreateSpace Independent Publishing Platform; 1st. ed., 2013; ISBN: 978-1492339243
- Como pensar como um Cientista da Computação usando Python (traduzido). Allen Downey, Jeffrey Elkner, and Chris Meyers. 2002.
- Python para Desenvolvedores. Luiz Eduardo Borges. Rio de Janeiro; 2010; 978-85-909451-1-6.



- Introdução à Programação com Python. Nilo Ney Coutinho, 2ª edição, 2014, ISBN: 978-85-7522-408-3.
- Learning to Program Using Python. Cody Jackson.
- CreateSpace Independent Publishing Platform; ISBN: 9781461182054

2. Complementar:

- Fundamentos da Programação de Computadores. Ana Fernanda Gomes Ascencio / Edilene Aparecida Veneruchi De Campos. 3ª edição; 2012, Pearson; ISBN 978-8564574168

Cidade Universitária “Prof. José Aloísio de Campos”, 26 de Abril de 2019.

Alberto Costa Neto
Professor da Disciplina

Kalil Araújo Bispo
Professor da Disciplina