



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ - ARA
CURSO DE TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO
PLANO DE ENSINO

SEMESTRE 2018.1

I. IDENTIFICAÇÃO DA DISCIPLINA:

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS		TOTAL DE HORAS-AULA SEMESTRAIS
		TEÓRICAS	PRÁTICAS	
CIT7580	Algoritmos e Programação	6		108

HORÁRIO		MODALIDADE
TURMAS TEÓRICAS	TURMAS PRÁTICAS	Presencial

II. PROFESSOR(ES) MINISTRANTE(S)

Prof. Lucas Borges Castellan
Email:lucas.castellan@ufsc.br

III. PRÉ-REQUISITO(S)

CÓDIGO	NOME DA DISCIPLINA
-	Esta disciplina não possui pré-requisitos

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

GRADUAÇÃO EM TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

V. JUSTIFICATIVA

Esta disciplina de caráter técnico prepara o discente para o desenvolvimento de soluções computacionais usando técnicas de programação. Além dos aspectos fundamentais da lógica de programação, esta disciplina também foca o uso de uma primeira linguagem de programação.

VI. EMENTA

Conceito e estrutura de algoritmo. Pseudo-código e fluxograma. Paradigma de programação estruturado x orientado a objetos. Linguagem Compilada x Interpretada. Noções de lógica de programação. Dados, expressões e algoritmos sequenciais. Comandos de entrada e saída, estruturas de controle de fluxo, operadores lógicos e aritméticos, estruturas de dados homogêneas e heterogêneas. Tipos definidos pelo usuário. Modularização. Introdução à uma Linguagem de Programação de alto nível. Atividades em laboratório com a linguagem selecionada.

VII. OBJETIVOS

Objetivo Geral:

Tornar os alunos capazes de visualizar soluções computacionais para problemas através da aplicação dos conceitos da lógica de programação e dotá-los da capacidade de construção de programas, em linguagem de alto nível estruturada, que implementem as soluções vislumbradas.

Objetivos Específicos:

- Desenvolver o raciocínio lógico e abstrato do aluno;
- Familiarizar o aluno com o modelo sequencial de computação;
- Apresentar técnicas e linguagens para representação e construção de algoritmos simples;
- Apresentar conceitos básicos de linguagens de programação;
- Treinar o aluno no processo básico de desenvolvimento de software concepção, edição, execução e teste de programas de computador);
- Capacitar o aluno no uso de uma linguagem de alto nível.

VIII. CONTEÚDO PROGRAMÁTICO

Conteúdo Teórico seguido de Conteúdo Prático com desenvolvimento de problemas em computador:

UNIDADE 1: Introdução [16 horas-aula]

- Conceito
- Algoritmo
- Funcionalidade de um algoritmo
- Estrutura de um algoritmo
- Pseudo-código
- Introdução a algoritmos em alto nível
- Classificação das linguagens de programação com relação à similaridade com a linguagem natural
 - linguagem de máquina
 - linguagem simbólica
 - linguagem de alto nível
- Exemplos de algoritmos

UNIDADE 2: Conceituação de elementos básicos para construção de um algoritmo [16 horas-aula]

- Constante
- Variável
- Identificador
- Palavra-reservada
- Operadores aritméticos, de atribuição, relacionais e lógicos
- Parâmetros
- Tipos de dados primitivos
 - Lógico
 - Caractere
 - Inteiro
 - Real
- Conceito de lógica
- Método para construção de um algoritmo
- Estrutura de um pseudocódigo
- Construção de algoritmos em pseudocódigo
- Estrutura de E/S de dados – teclado e monitor

UNIDADE 3: Estruturas de controle de fluxo: seleção [16 horas-aula]

- Estruturas de seleção
 - Seleção simples
 - Seleção composta

UNIDADE 4: Estruturas de controle de fluxo: repetição [16 horas-aula]

- Estruturas de repetição
 - enquanto faça
 - para faça

UNIDADE 5: Variáveis compostas [12 horas-aula]

- Vetores unidimensionais e multidimensionais
 - caracterização, declaração e indexação

UNIDADE 6: Conceitos básicos de Linguagens de Programação [16 horas-aula]

- Conceituação de Linguagem de Programação
- Atividades de programação com uma linguagem de programação
- Codificação, compilação/interpretação e execução

UNIDADE 7: Modularização [16 horas-aula]

- Modularização
 - Definição de módulos
 - Procedimentos/Funções/Métodos
 - Parâmetros (por valor e referência)

IX. METODOLOGIA DE ENSINO / DESENVOLVIMENTO DO PROGRAMA

1. Aulas expositivas intercaladas com discussões. Material de apoio postado no Moodle. Desenvolvimento de trabalhos e exercícios;
2. Atividades práticas no computador, utilizando o ambiente de desenvolvimento IDLE e a Linguagem de Programação Python.

Requisitos de infraestrutura necessários para ministrar as aulas:

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

X. 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 três avaliações, sendo:
 - **AV1:** Avaliação Escrita 1 - referente a resolução de problemas utilizando algoritmos
 - **AV2:** Avaliação Escrita 2 - referente a resolução de problemas utilizando algoritmos e a linguagem de programação Python.
 - **MT:** Trabalho Prático e Lista de Exercícios

- A Média Final (MF) será calculada da seguinte forma:

$$MF = [(AV1 + AV2) / 2] * 0,7 + MT * 0,3$$

- A nota mínima para aprovação na disciplina será $MF \geq 6,0$ (seis) e Frequência Suficiente (FS). (Art. 69 e 72 da Res. nº 17/Cun/1997).
- A **apresentação** do trabalho prático é **obrigatória**. A avaliação do trabalho prático é feita individualmente, mesmo que o trabalho seja feito em grupo. Desta forma, caso o aluno não apresente o trabalho, a nota **MT** é igual a 0 (zero).
- 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 comparecer às avaliações ou não apresentar trabalhos no prazo estabelecido será atribuída nota 0 (zero). (Art. 70, § 4º da Res. nº 17/CUn/1997)

Observações:

Avaliação de recuperação

- Não há avaliação de recuperação nas disciplinas de **caráter prático** que envolve atividades de laboratório (Res.17/CUn/97).

Avaliação de segunda chamada:

- Para pedido de segunda avaliação somente em casos em que o aluno, por motivo de força maior e plenamente justificado, deixar de realizar avaliações previstas no plano de ensino, deverá formalizar pedido de avaliação à coordenação do curso dentro do prazo de 3 dias úteis apresentando comprovação.

XI. CRONOGRAMA TEÓRICO/PRÁTICO

AULA (semana)	DATA	ASSUNTO
1	26/02/2018 à 03/03/2018	Apresentação do plano de ensino e da disciplina Unidade 1: Introdução à arquitetura de computadores: memória, processador, bits e bytes Introdução a algoritmos
2	04/03/2018 à 10/03/2018	Estrutura de um algoritmo Pseudo-código Introdução a algoritmos em alto nível Exemplos e exercícios

		Unidade 2: Visão geral das linguagens de programação Declaração de variáveis.
3	11/03/2018 à 17/03/2018	Palavra-reservada Operadores aritméticos, de atribuição, relacionais e lógicos Parâmetros Tipos de dados primitivos Método para construção de um algoritmo Estrutura de um pseudocódigo Construção de algoritmos em pseudocódigo Estrutura de E/S de dados – teclado e monitor Exemplos e exercícios
4	18/03/2018 à 24/03/2018	Unidade 3: Controle de Fluxo: seleção Estrutura de seleção simples, composta e encadeada e exercícios. Estrutura de seleção de múltipla escolha e exercícios. Exemplos e exercícios
5	25/03/2018 à 31/03/2018	Estrutura de seleção de múltipla escolha e exercícios. Exemplos e exercícios Unidade 4: Controle de Fluxo: repetição Estruturas de repetição
6	01/04/2018 à 07/04/2018	Estruturas de repetição Enquanto-faça Para-faça Exemplos e exercícios
7	08/04/2018 à 14/04/2018	Estruturas de repetição Enquanto-faça Para-faça Exemplos e exercícios Disponibilização dos temas dos trabalhos
8	15/04/2018 à 21/04/2018	Estruturas de repetição (Exercícios) Revisão (Estrutura de repetição) Avaliação I
9	22/04/2018 à 28/04/2018	Unidade 5: Variáveis compostas Vetores unidimensionais Exercícios Feriado (04/05-Quinta-Feira)
10	29/04/2018 à 05/05/2018	Vetores unidimensionais (Exercícios) Vetores multidimensionais Exercícios
11	06/05/2018 à 12/05/2018	Unidade 6: Conceitos básicos de Linguagens de Programação Conceituação de Linguagem de Programação Classificação das linguagens de programação com relação à similaridade com a linguagem natural
12	13/05/2018 à 19/05/2018	Introdução ao ambiente de desenvolvimento IDLE - Codificação Atividades de programação com uma linguagem de programação Exemplos e exercícios
13	20/05/2018 à 26/05/2018	Unidade 7: Modularização Definição de módulos Procedimentos/Funções/Métodos Exercícios
14	27/05/2018 à 02/06/2018	Parâmetros (por referência e valor) Exercícios Introdução ao Turtle, características, funções
15	03/06/2018 à 09/06/2018	Avaliação Feriado
16	10/06/2018 à 16/06/2018	Entrega dos Trabalhos no AVA Apresentação dos Trabalhos
17	17/06/2018 à 23/06/2018	Prova substitutiva (solicitação junto a secretária) Nova Avaliação (Prova de recuperação)
18	24/06/2018 à 30/06/2018	Publicação de Notas

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

XII. Feriados previstos para o semestre 2018.1:

DATA	Feriados/Dias não letivos
30/03	Feriado Nacional – 6ª feira Santa
31/03	Dia não letivo
03/04	Aniversário da cidade de Araranguá
21/04	Feriado Nacional – Dia de Tiradentes
30/04	Dia não letivo
01/05	Feriado Nacional – Dia do Trabalhador
04/05	Dia da Padroeira da cidade de Araranguá
31/05	Feriado Nacional – <i>Corpus Christi</i>

XIII. BIBLIOGRAFIA BÁSICA

FORBELLONE, André L. V.; EBERSPACHER, Henri F. Lógica de Programação. 3. ed. Pearson, 2005.

MCGUGAN, Will. Beginning Game Development with Python and Pygame: From Novice to Professional. Berkeley, CA: Apress, Inc., 2007. ISBN 9781430203254 Disponível em : <<http://dx.doi.org/10.1007/978-1-4302-0325-4>>. Acesso em : 9 out. 2009.

MENEZES, Nilo Ney Coutinho. Introdução à programação com Python: algoritmos e lógica de programação para iniciantes. São Paulo: Novatec, 2010. 222 p. ISBN 9788575222508.

XIV. BIBLIOGRAFIA COMPLEMENTAR:

ARAÚJO, Everton Coimbra de. Algoritmos Fundamentos e Prática. Visual Books, 2007.

HETLAND, Magnus Lie. Beginning Python: From Novice to Professional. Second Edition. Berkeley, CA: Apress, 2008. ISBN9781430206347 Disponível em : <<http://dx.doi.org/10.1007/978-1-4302-0634-7>>. Acesso em : 9 out. 2009.

MEDINA, Marco; FERTIG, Cristina. Algoritmos e Programação – Teoria e Prática. 2. ed.. São Paulo: Novatec, 2006.

LANGTANGEN, Hans Petter. Python Scripting for Computational Science. Third Edition. Berlin: Springer-Verlag Berlin Heidelberg, 2008. (Texts in Computational Science and Engineering, 1611-0994; 3).

XAVIER, Gley Fabiano Cardoso. Lógica de programação. 11. ed. São Paulo (SP): SENAC São Paulo, 2007.

LOPES, Anita; GARCIA, Guto. Introdução à programação: 500 algoritmos resolvidos. Rio de Janeiro: Campus, 2002. 469 p. ISBN 8535210199.

Os livros acima citados constam na Biblioteca Universitária e Setorial de Araranguá. Algumas bibliografias também podem ser encontradas na Biblioteca Virtual da UFSC.

Lucas Borges Castellan

/ / 2018

Aprovado pelo
departamento em

/ / 2018


Aprovado pelo colegiado do curso
de graduação em

/ / 2018

