



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ - ARA

PLANO DE ENSINO

SEMESTRE 2016.2

I. IDENTIFICAÇÃO DA DISCIPLINA:

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS		TOTAL DE HORAS-AULA SEMESTRAIS
		TEÓRICAS	PRÁTICAS	
ARA7140	Programação em Computadores I	-	4	72

TURMAS TEÓRICAS	HORÁRIO	MODALIDADE
	TURMAS PRÁTICAS	Presencial
	01652B 4.1830-2 e 5.1830-2	

II. PROFESSOR(ES) MINISTRANTE(S)

Prof. Jailson Torquato
E-mail: jailson.torquato

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

Bacharelado 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 de algoritmo. Pseudo-código e fluxograma. Estrutura de um algoritmo. identificadores, palavras reservadas, variáveis, constantes, declaração de variáveis, comandos de entrada e saída, estruturas de controle de fluxo, estruturas de dados homogêneas (vetores e matrizes) e heterogêneas (registros).

Tipos definidos pelo usuário. Modularização. Introdução à uma Linguagem de Programação de alto nível estruturada. 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 [8 horas-aula]

- Conceito
- Algoritmo
- Funcionalidade de um algoritmo
- Exemplos de algoritmos

UNIDADE 2: Conceitos básicos de Linguagens de Programação [4 horas-aula]

- Conceituação de Linguagem de Programação
- Classificação das linguagens de programação com relação à similaridade com a linguagem natural
 - linguagem de máquina
 - linguagem simbólica
 - linguagem de alto nível
- Atividades de programação com uma linguagem de programação
- Codificação, compilação/interpretação e execução

UNIDADE 3: Conceituação de elementos básicos para construção de um algoritmo [8 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 4: Estruturas de controle de fluxo: seleção [16 horas-aula]

- Estruturas de seleção
 - Seleção simples
 - Seleção composta

UNIDADE 5: Estruturas de controle de fluxo: repetição [16 horas-aula]

- Estruturas de repetição
 - enquanto faça
 - para faça

UNIDADE 6: Variáveis compostas [12 horas-aula]

- Vetores unidimensionais e multidimensionais
 - caracterização, declaração e indexação
- Listas e Strings
 - caracterização, declaração e indexação

UNIDADE 7: Modularização [8 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 e a linguagem de programação Python.
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.

Horário de atendimento ao aluno: .

- Quarta-feira 16:00 – 17:00. Jardim das Avenidas - Sala 307
- Quinta-feira 16:00 – 18:00. Jardim das Avenidas – Sala 307

XI. CRONOGRAMA TEÓRICO/PRÁTICO

AULA (semana)	DATA	ASSUNTO
1	08/08 a 12/08/16	Apresentação do plano de ensino e da disciplinas Unidade 1: Introdução à arquitetura de computadores: memória, processador, bits e bytes Introdução a algoritmos
2	15/08 a 19/08/16	Estrutura de um algoritmo Pseudo-código Introdução a algoritmos em alto nível Introdução ao ambiente de desenvolvimento IDLE Exemplos e exercícios

3	22/08 a 26/08/16	Unidade 2: Conceitos básicos de Linguagens de Programação Conceituação de Linguagem de Programação Classificação das linguagens de programação com relação à similaridade com a linguagem natural Codificação, compilação/interpretação e execução
4	29/08 a 02/09/16	Unidade 3: Visão geral das linguagens de programação Declaração de variáveis. Chamadas de Funções Tipos de variáveis. Operador de atribuição. Resolução de exercícios. Estrutura de E/S de dados Exemplos e exercícios
5	05/09 a 09/09/16	Feriado (07/09) Estrutura de E/S de dados Operadores: aritméticos, lógicos e relacionais. Exemplos e exercícios Unidade 4: Controle de Fluxo: seleção Introdução ao Turtle, características, funções Estrutura de seleção simples, composta e encadeada e exercícios.
6	12/09 a 16/09/16	II Semana Acadêmica de Tecnologias da Informação e Comunicação
7	19/09 a 23/09/16	Unidade 4: Controle de Fluxo: seleção Estrutura de seleção simples, composta e encadeada e exercícios. Estrutura de seleção de múltipla escolha e exercícios. Exemplos e exercícios
8	26/09 a 30/09/16	Estrutura de seleção simples, composta e encadeada e exercícios. Estrutura de seleção de múltipla escolha e exercícios.
9	03/10 a 07/10/16	Estrutura de seleção Disponibilização dos temas dos trabalhos Revisão Avaliação I (06/10)
10	10/10 a 14/10/16	Feriado (12/10) Unidade 5: Controle de Fluxo: repetição Estruturas de repetição Exercícios
11	17/10 a 21/10/16	Estruturas de repetição Exercícios
12	24/10 a 28/10/16	Estruturas de repetição Exercícios Unidade 6: Vetores, Listas e Strings Listas: Operações e métodos Listas: cadeia de caracteres (String) Exercícios
13	31/10 a 04/11/16	Feriado (02/11) Exercícios
14	07/11 a 11/11/16	Unidade 7: Modularização Definição de módulos Procedimentos/Funções/Métodos Parâmetros (por referência e valor) Exercícios
15	14/11 a 18/11/16	Entrega dos Trabalhos no AVA (15/11)
16	21/11 a 25/11/16	Apresentação dos Trabalhos (16/11 e 17/11) Avaliação II (24/11)
17	28/11 a 02/12/16	Nova Avaliação (Prova de recuperação): Prova Prática compreendendo todo o conteúdo da disciplina (01/12)
18	05/12 a 09/12/16	Publicação de Notas

II. Feriados previstos para o semestre 2016.2

07/09/2016

Independência do Brasil

12/10/2016	Nossa Senhora Aparecida
28/10/2016	Dia do Servidor Público (Lei 8112 art.236)
02/11/2016	Finados
14/11/2016	Dia não letivo
15/11/2016	Proclamação da República

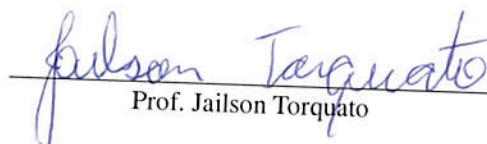
XIII. BIBLIOGRAFIA BÁSICA

- HETLAND, Magnus Lie. **Beginning Python: From Novice to Professional**. Second Edition. Berkeley, CA: Apress, 2008. ISBN 9781430206347 Disponível em : <<http://dx.doi.org/10.1007/978-1-4302-0634-7>>. Acesso em : 9 out. 2009.
- 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.

XIV. BIBLIOGRAFIA COMPLEMENTAR:

- ARAÚJO, Everton Coimbra de. **Algoritmos Fundamentos e Prática**. Visual Books, 2007.
- 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 no acervo da disciplina, via sistema Moodle.


Prof. Jailson Torquato


Prof.ª Patricia Jantsch Fiuza
Coordenadora do Curso

Prof.ª. Patricia Jantsch Fiuza
Coordenadora do Curso de
Tecnologias da Informação e Comunicação
Portaria 101/2015/GR
SIAPE: 2058903
UFSC Centro Araranguá

Aprovado na Reunião do Colegiado do Curso ___/___/___