



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE - CTS
DEPARTAMENTO DE COMPUTAÇÃO - DEC

PLANO DE ENSINO

SEMESTRE 2018.2

I. IDENTIFICAÇÃO DA DISCIPLINA:

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS		TOTAL DE HORAS-AULA SEMESTRAIS
		TEÓRICAS	PRÁTICAS	
CIT7141	Programação em Computadores II	0	4	72

HORÁRIO		MÓDULO
TURMAS TEÓRICAS	TURMAS PRÁTICAS	Presencial
	02652B 2.2020-2 e 4.1830-2	

II. PROFESSOR(ES) MINISTRANTE(S)

Miriam Z. Parra Sejas
miriamsejasz@gmail.com

III. PRÉ-REQUISITO(S)

CÓDIGO	NOME DA DISCIPLINA
CIT7140	Programação em Computadores I

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

Bacharelado em Tecnologias da Informação e Comunicação

V. JUSTIFICATIVA

Esta disciplina é necessária para o aprimoramento dos conceitos de programação em computadores utilizando uma linguagem de alto nível. Fornece subsídios adicionais para permitir o desenvolvimento de sistemas computacionais de maior complexidade.

VI. EMENTA

Ponteiros. Alocação dinâmica. Tipo abstrato de dado. Implementação de estruturas de dados alocadas dinamicamente. Recursividade. Manipulação de arquivos. Fundamentos de algoritmos de ordenação e pesquisa de dados. Uso de uma linguagem de programação em laboratório.

VII. OBJETIVOS

Objetivo Geral:

Proporcionar aos alunos conhecimentos avançados de programação em linguagem orientada a objetos de alto nível de modo que seja possível o desenvolvimento de sistemas complexos.

Objetivos Específicos:

- Aprimorar o raciocínio lógico e abstrato do aluno;
- Aprimorar o entendimento do aluno a respeito do modelo sequencial de computação;
- Apresentar técnicas para a construção de algoritmos complexos;
- Aprimorar os conceitos de linguagens de programação;
- Treinar e capacitar o aluno no processo de desenvolvimento de software.

VIII. CONTEÚDO PROGRAMÁTICO

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

UNIDADE 1: Introdução a Programação O.O. [12 ha]

- Introdução a Orientação a Objetos
- Classes e Objetos
- Tipos de dados

UNIDADE 2: Modularização [12 ha]

- Definição de módulos
- Métodos
- Escopo de variáveis
- Passagem de parâmetros
 - Por valor
 - Por referência
- Valor de retorno

UNIDADE 3: Alocação dinâmica de memória [04 ha]

- Funções de alocação e liberação de memória
- Alocação de vetores
- Alocação de matrizes
- Alocação de Objetos

UNIDADE 4: Tipo abstrato de dado [08 ha]

- Tipos enumerativos
- Estruturas não-homogêneas
- Classes e Objetos

UNIDADE 5: Estrutura de dados [12 ha]

- Alocação de estruturas de dados
- Exemplos de estruturas de dados
- Listas, Conjuntos e Tabelas Hash

UNIDADE 6: Recursividade [04 ha]

- Algoritmos recursivos
- Funções recursivas

UNIDADE 7: Manipulação de arquivos [08 ha]

- Arquivos textos e binários
- Abertura e fechamento
- Leitura e escrita

UNIDADE 8: Ordenação e pesquisa de dados [12 ha]

- Algoritmos elementares
- Algoritmos otimizados
- Busca em vetores e em estrutura de dados

IX. METODOLOGIA DE ENSINO / DESENVOLVIMENTO DO PROGRAMA

1. A disciplina será ministrada com aulas expositivas fornecendo os componentes teóricos. Material de apoio postado no Moodle. Desenvolvimento de trabalho e exercícios;
2. Atividades práticas no computador, utilizando ferramenta de desenvolvimento para a Linguagem Java.

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 duas avaliações escritas (AV):
 - **AV1**: Avaliação Escrita 1 - será referente aos conteúdos das Unidades 1 à 4;
 - **AV2**: Avaliação Escrita 2 - será referente aos conteúdos das Unidades 1 à 8;
- A Média das avaliações **AV1** e **AV2** terá peso 7.
- A Média de trabalhos (MT) será referente aos Trabalhos de programação em Linguagem Java (peso 3)

- 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 >= 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)
- Caso seja encontrado **Cópia(s)** e/ou **Plágio(s)** em **qualquer avaliação**, seja em avaliação individual (AV1 e AV2) ou nos trabalhos e listas de exercícios (MT), o aluno estará automaticamente reprovado como nota ZERO (caso não esteja reprovado por FI)
- 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).

Nova avaliação

- Pedidos de segunda avaliação somente para casos em que o aluno, por motivo de força maior e plenamente justificado, deixar de realizar avaliações previstas no plano de ensino, e deverá ser formalizado via requerimento de avaliação à Secretaria Acadêmica do Campus Araranguá dentro do prazo de 3 dias úteis apresentando comprovação. (Ver formulário)

XI. CRONOGRAMA PRÁTICO		
AULA (semana)	DATA	ASSUNTO
1	30/07 a 04/08	Apresentação do plano de ensino e visão geral de programação II. Atividades em laboratório e teóricas. UNIDADE 1: Introdução a Programação O.O.
2	06/08 a 11/08	Classes e Objetos Tipos de Dados Exercícios
3	13/08 a 18/08	Classes e Objetos Tipos de Dados Exercícios
4	20/08 a 25/08	UNIDADE 2: Modularização, atividade prática, resolução de problemas, compilação e execução Uso de ferramentas de debug para verificar o funcionamento de um programa/encontrar falhas
5	27/08 a 01/09	Métodos Escopo de variáveis Parâmetros - passagem de valores e referência Exercícios
6	03/09 a 08/09	Exercícios
7	10/09 a 15/09	UNIDADE 3: Funções de alocação e liberação de memória Alocação de vetores
8	17/09 a 22/09	Alocação de matrizes Alocação de Objetos UNIDADE 4: Tipos enumerativos Estruturas não-homogêneas Classes e Objetos
9	24/09 a 29/09	Estruturas não-homogêneas Classes e Objetos Exercícios
10	01/10 a 06/10	Avaliação 1 UNIDADE 5: Alocação de estruturas de dados Exercícios
11	08/10 a 13/10	Exemplos de estruturas de dados Listas, Conjuntos e Tabelas Hash Exercícios
12	15/10 a 20/10	Exemplos de estruturas de dados Listas, Conjuntos e Tabelas Hash Exercícios UNIDADE 6: Manipulação de arquivos Exercícios
13	22/10 a 27/10	Manipulação de arquivos Exercícios UNIDADE 7: Ordenação e pesquisa de dados Exercícios
14	29/10 a 03/11	Ordenação e pesquisa de dados Exercícios
15	05/11 a 10/11	Ordenação e pesquisa de dados Exercícios
16	12/11 a 17/11	Exercícios Avaliação 2

17	19/11 a 24/11	Prova substitutiva Nova Avaliação (Prova de recuperação) Prova Prática compreendendo todo o conteúdo da disciplina
18	26/11 a 01/12	Divulgação de Notas

XII. Feriados previstos para o semestre 2018.1:

DATA	
07/09/2018	Independência do Brasil (Sexta)
08/09/2018	Dia não letivo (Sábado)
12/10/2018	Nossa Senhora Aparecida (Sexta)
13/10/2018	Dia não letivo (Sábado)
28/10/2018	Dia do Servidor Público (Domingo)
02/11/2018	Finados (Sexta)
03/11/2018	Dia não letivo (Sábado)
15/11/2018	Proclamação da República (Quinta)
16/11/2018	Dia não letivo (Sexta)
17/11/2018	Dia não letivo (Sábado)

XIII. BIBLIOGRAFIA BÁSICA

CORMEN, Thomas et al. **Algoritmos: teoria e prática**. Rio de Janeiro: Campus, 2002.

GOODRICH, Michael T.; TAMASSIA, Roberto. **Estruturas de dados e algoritmos em Java**. 4. ed. Porto Alegre: Bookman, 2007 xiii, 600 p. ISBN 9788560031504.

SANTOS, Rafael. **Introdução à programação orientada a objetos usando JAVA**. Rio de Janeiro: Campus, 2003. 3 19p. ISBN 853521206X.

XIV. BIBLIOGRAFIA COMPLEMENTAR:

DEITEL, Harvey M.; DEITEL, Paul J. **Java como programar**. 8. ed. São Paulo: Pearson Prentice Hall, 2010. xxix, 1144 p. ISBN 9788576055631.

PUGA, Sandra; RISSETTI, Gerson. **Lógica de programação e estruturas de dados: com aplicações em Java**. 2. ed. São Paulo: Prentice Hall, 2009. 262 p. ISBN 9788576052074.

ZIVIANI, Nivio. **Projeto de algoritmos: com implementações em Java e C++**. São Paulo: Cengage Learning, c2007. xx, 621 p. ISBN 9788522105250.

ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ e Java**. 2. ed. São Paulo: Pearson Prentice Hall, 2008. 434 p. ISBN 9788576051480.

KNUTH, Donald E. **Art of Computer Programming: Sorting and Searching**. 2. ed. Addison-Wesley Professional, vo1. 3. 1998.

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.

Miriam

Zareth Parra

Sejas:704311

61127

Assinado de forma
digital por Miriam
Zareth Parra

Sejas:70431161127

Dados: 2018.06.07

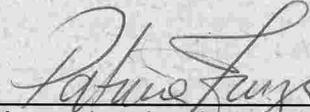
23:50:43 -03'00'

Prof. Miriam Zareth Parra Sejas
Professor da Disciplina

30/05 / 2018

Aprovado pelo departamento em

/ / 2018



Aprovado pelo colegiado do curso
de graduação em

/ / 2018