



ANEXO DA RESOLUÇÃO Nº 30/2011, DO CONSELHO DE GRADUAÇÃO

Faculdade de Matemática

COLEGIADO DO CURSO DE ESTATÍSTICA

PLANO DE ENSINO

1. IDENTIFICAÇÃO

COMPONENTE CURRICULAR: Introdução à Computação 1				
UNIDADE OFERTANTE: Faculdade de Computação				
CÓDIGO: GES005		PERÍODO: 2º		TURMA: E
CARGA HORÁRIA			NATUREZA	
TEÓRICA: 60	PRÁTICA: 00	TOTAL: 60	OBRIGATÓRIA: (X)	OPTATIVA: ()
PROFESSOR: Anilton Joaquim da Silva				ANO: 2018/1
OBSERVAÇÕES:				

2. EMENTA

Noções Fundamentais: aplicação de computadores; computador (e seus componentes); arquitetura de von Neumann e arquiteturas alternativas; programa de computador; compilador/interpretador; linguagem de programação; sistema operacional; representação binária (sinal-magnitude e complemento-de-dois); conversão entre sistemas numéricos; operações binárias (lógica e aritmética). Algoritmo: descrição de algoritmos; raciocínio para construção de algoritmos. Programação: organização de um programa-fonte em uma linguagem procedimental; operações de atribuição, de iteração, de entrada/saída, de formatação de saída. Depuração: utilização de ferramentas para detectar erros em programa.

3. JUSTIFICATIVA

A resolução de problemas de forma lógica e metódica é essencial em um estatístico. O estudo da programação de computadores estimula, pelo exercício de desenvolvimento de algoritmos, o raciocínio lógico no estudante de estatística. Além disso, muitos dos algoritmos da estatística, matemática e ciências, mesmo que conceitualmente simples, demandam um volume considerável de trabalho para sua execução manual. A programação de computadores permite a automatização destes algoritmos e torna possível a solução de problemas recorrentes na estatística de forma mais eficiente.

4. OBJETIVO

Objetivo Geral:

- Construir um programa, em linguagem procedimental, que apresente solução para um problema computacionalmente solúvel.



Objetivos Específicos:

- Discutir sobre a utilidade dos computadores digitais na solução de problemas cotidianos.
- Discutir sobre os principais componentes (em hardware e software) de um computador (com arquitetura de von Neumann).
- Representar valores inteiros (positivos e negativos) em binário (sinal-magnitude e complemento-de-dois).
- Realizar operações de conversão de sistema numérico: decimal-binário, decimal-hexadecimal, binário-hexadecimal.
- Realizar operações binárias: lógica (OU e E), aritmética (Soma).
- Construir algoritmos para a solução de um dado problema a partir de diferentes ferramentas: fluxograma, pseudocódigo.
- Escrever um programa correspondente a um algoritmo.
- Gerar código executável a partir de um programa-fonte.
- Encontrar erros de um programa por meio de ferramentas de depuração.

5. PROGRAMA

1. Noções Básicas

- 1.1 computadores como instrumento do dia a dia: histórico, aplicações e benefícios.
- 1.2 Placa mãe, e dispositivos de: armazenamentos, entrada e saída (arquitetura Von Neuman).
- 1.3 Terminologia: (hardware, software, programa, bit, byte, códigos: Binário e ASCII).
- 1.4 Representação binária
 - 1.4.1 Sistemas numéricos: bases 2, 10 e 16, e operações básicas (conversão entre as bases, e soma de binários).
 - 1.4.2 Representação binária: sinal-magnitude e complemento-de-dois.
- 1.5 Principais softwares em um computador: Sistema operacional, interpretador/compilador.

2. Introdução a Algoritmos

- 2.1 Abstração: representação do mundo real no computador – algoritmo (pseudocódigo, fluxograma, linguagem algorítmica, e C .
- 2.2 Tipos primitivos de dados e variáveis.
- 2.3 Expressões: aritméticas, relacionais e lógicas.
- 2.4 Comandos de: atribuição, entrada e saída de dados.
- 2.5 Estruturas de controle de fluxo: seleção e repetição.
- 2.6 Exemplos de algoritmos - Uso da IDE VisuAlg (digitar e interpretar algoritmos).

3. Introdução à Linguagem de Programação C

- 3.1 A estrutura de programas em C (main(), e diretivas para o compilador).
- 3.2 Uso da IDE CodeBlock: digitar, compilar, e executar programas em C.
- 3.3 Tipos de dados primitivos: char, int, float.
- 3.4 Modificadores: unsigned, long.
- 3.5 Entrada/saída padrão: stdin, stdout.
- 3.6 Comandos de entrada/saída: scanf, printf, com formato: %d, %f, e %c.

4. Estrutura Sequencial - comando de atribuição

- 4.1 Expressões aritméticas: operadores e construções especiais.
- 4.2 Exemplos.



5. Depuração: uso de ferramentas de depuração para detectar erros em programas
 - 5.1 Pontos de parada - Breakpoints.
 - 5.2 Apresentação dos conteúdos de variáveis - display.
 - 5.3 Execuções: run, step e next (passo-a-passo).
 - 5.4 Exemplos.
6. Estruturas de Seleção (condição)
 - 6.1 Expressões relacional/lógica: dados (true (1), false (0)), e operadores relacional/lógica.
 - 6.2 Comando if - else.
 - 6.3 Comando switch – case.
 - 6.4 Exemplos.
7. Estruturas de Repetição (iteração)
 - 7.1 Comando while.
 - 7.2 Comando do ... while.
 - 7.3 Comando for.
 - 7.4 Exemplos.
8. Variável - Vetor de int e Vetor de float
 - 8.1 Conceitos sobre vetor: porque e como usar vetores.
 - 8.2 Endereçamento de uma posição de um vetor.
 - 8.3 Varreduras, leitura e impressão de vetores.
 - 8.4 Exemplos.
9. Variável - Vetor de char
 - 9.1 Conceitos sobre cadeia de caracteres: string, \0.
 - 9.2 Leitura e impressão de strings: scanf, printf, com formato %s e %[].
 - 9.3 Usos de funções relacionadas à string: strlen, strcmp, strcpy, e atoi.
 - 9.4 Exemplos.
10. Variável - vetor de vetor – Matrizes de int, float e char
 - 10.1 Conceitos sobre matrizes.
 - 10.2 O uso de matrizes em C: endereçamento de uma posição, entrada/saída.
 - 10.3 Exemplos.
11. Entrada/saída padrão: stderr
 - 11.1 A diferença entre usar printf e fprintf + stderr para mensagem de erro de programa.
 - 11.2 Exemplos.

6. METODOLOGIA

O curso será composto por aulas teóricas e suas práticas. Nas aulas teóricas serão realizadas aulas expositivas utilizando *data-show*, complementadas com listas de exercícios para fixação do conhecimento. As práticas, aulas de exercícios, serão ministradas em laboratórios, onde o aluno terá contato direto com o computador e com os aplicativos utilizados no curso. Estas aulas serão ministradas através de estudos dirigidos, complementados com exercícios. Além disso, serão disponibilizados monitores para auxiliar o professor no esclarecimento de dúvidas que possam surgir durante as aulas práticas ou durante horários de



atendimento previamente definidos, bem como aplicar estratégias de recuperação para os alunos de menor rendimento (atividades de reforços).

CRONOGRAMA:

MÓDULOS	MÊS
1 e 2	março
3, 4, 5 e 6	abril
7 e 8	maio
9 e 10	Junho e julho
11	julho

7. AVALIAÇÃO

A avaliação do aluno será feita por meio de listas de exercícios (trabalhos), provas práticas individuais, provas dissertativas individuais:

Listas de exercícios, em todos os módulos - 5 pontos;

Prática: Estudos dirigidos, dos módulos 2, 3, 4, 5, 6, 7, 8, 9 e 10 - 20 pontos;

Prova individual dos módulos 2, 3, 4, e 6 - 25 pontos;

Prova individual dos módulos 7 e 8 - 25 pontos;

Prova individual dos módulos 9 e 10 - 25 pontos;

Está previsto também 1 prova substitutiva (25 pontos) aos alunos que não obtiverem nota suficiente para aprovação. A recuperação de aprendizagem será feita por meio de atendimento individual (ou em grupo) realizado pelo professor em horário extraclasse. A recuperação de aprendizagem também será realizada pela aplicação de exercícios extras e uso de material complementar para estudo dirigido.

8. BIBLIOGRAFIA

Básica

ASCENCIO, A. F.; CAMPOS, E. A.V. de. **Fundamentos da Programação de Computadores**. Editora Prentice Hall, São Paulo, 2007.

MIZRAHI, V. V. **Treinamento em linguagem C**. Editora Pearson Prentice Hall, 2010.

BACKES, A. R. **Linguagem C: Completa e Descomplicada**, ELSEVIER, 2013.



Complementar

CAPRON, H. L.; JOHNSON, J. A. **Introdução à Informática**. Pearson Prentice Hall, São Paulo, 2004.

MOKARZEL, F. C.; SOMA, N. Y. **Introdução à Ciência da Computação**. Editora Elsevier-Campus, 2008.

FORBELLONE, A. L. V.; EBERSPÖCHER, H. F. **Lógica de Programação: A Construção de Algoritmos e Estruturas de Dados**, Editora Pearson Prentice Hall. 2005.

GUIMARÃES, A.M.; LAGES, N.A.C. **Introdução à Ciência da Computação**. LTC Ed. S.A., 1985.

SCHILDT, H. C. **Completo e Total**. Editora Pearson-Makron Books., 1997.

KERNIGHAM, B. & RITCHIE, D. C a linguagem de programação: padrão ANSI. Editora Campus, 1990.

FEDELI, R. D., POLLONI, E. G. F., PERES, F. E. **Introdução à ciência da computação**. 2a. ed., São Paulo, Editora Cengage Learning, 2010.

"

9. APROVAÇÃO

Aprovado em reunião do Colegiado realizada em: ____/____/____

Coordenação do Curso de Graduação em: _____