



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE MATEMÁTICA
CURSO DE BACHARELADO EM ESTATÍSTICA

FICHA DE COMPONENTE CURRICULAR

CÓDIGO: GES009	COMPONENTE CURRICULAR: INTRODUÇÃO À COMPUTAÇÃO 2	
UNIDADE ACADÊMICA OFERTANTE: FACULDADE DE COMPUTAÇÃO		SIGLA: FACOM
CH TOTAL TEÓRICA: 60	CH TOTAL PRÁTICA: 0	CH TOTAL: 60

OBJETIVOS

Ensinar o discente a construir programa, em linguagem procedimental, de forma modular, que apresente solução para um problema computacionalmente solúvel; organizar o programa-fonte em várias funções fortemente coesas e fracamente acopladas; criar programas que façam uso de tipos de dados complexos; criar programas que realizam alocação dinâmica de variáveis; criar e compilar biblioteca de funções e gerar código objeto para as mesmas; gerar código executável a partir do código objeto de várias bibliotecas; utilizar alocação dinâmica de endereços de memória; detectar erro de programas (com várias funções) por meio de ferramenta de depuração.

EMENTA

Função: argumento e parâmetro; passagem de argumentos (valor e endereço); criação e chamada de funções; organização de um programa em várias funções; coesão e acoplamento em funções. Tipo de dado complexo: tipo de dado criado pelo programador; tipo de dado composto por outros tipos de dados; declaração e uso de variáveis de tipo complexo. Biblioteca: criação de uma biblioteca; compilação e geração de código objeto para uma biblioteca. Alocação dinâmica de endereços de memória: endereços de memória (alocação estática e dinâmica); declaração e uso de variáveis do tipo ponteiro; aritmética de ponteiros; uso de ponteiros para variáveis de tipo complexo. Depuração: utilização de ferramentas para detectar erros em programa que usam bibliotecas e variáveis do tipo ponteiro.

PROGRAMA

1. Função

- O que são argumentos e parâmetros.
- Passagem de argumentos (valor e endereço).
- Resultado retornado por uma função
 - Uso do resultado de retorno de uma função no programa.
- Criação e chamada de funções

- Variáveis locais e globais;
- Escopo de variáveis;
- Declaração de biblioteca criada pelo programador
 - **#include** “<nome-da-biblioteca>”
- Parâmetros da função **main ()**: **argc** e **argv**
 - Passagens de argumentos para um programa em tempo de lançamento do mesmo.
- Organização de um programa em várias funções
 - Modularização de programas;
 - Aplicação dos conceitos de coesão e acoplamento.

2. Manipulação de arquivos

- O que é um arquivo para um programa em C
 - Arquivos-padrão: **stdin, stdout, stderr**
 - Variável descritor de arquivo: **FILE ***
 - Dados apontados pelo ponteiro do arquivo
 - Tamanho do arquivo, endereço de início, *buffer* de entrada/saída
- Funções para manipulação com arquivos
 - **fopen, fclose, fgetc, fputc, fscanf, fprintf, fgetpos, fseek, fsetpos, ftell, rewind**
 - Valores retornados pelas funções acima

3. Tipo de dado complexo

- Declaração de uma **struct**
 - **Struct** composta por membros de tipos básicos;
 - **Struct** composta por membros de tipos básicos, vetores e por outras structs.
- Definição de uma variável tipo **struct**
 - Sintaxe de acesso aos membros da **struct**.
- Criação de um tipo não padrão: **typedef**
 - Definição de variáveis com tipos criados por **typedef**.
- Passagem de argumentos para funções cujos parâmetros são **structs**.

4. Biblioteca

- O que é uma biblioteca
 - Bibliotecas padrões do GCC: **stdio.h, string.h, stdlib.h**
- Criação de uma biblioteca
 - Encapsulamento;
 - Reutilização;
 - Manutenibilidade de programas que usam bibliotecas.
- Compilação e geração de código objeto para uma biblioteca
 - O que é o código executável de um programa;
 - O que é um código-objeto (relocável);
 - Geração do código-objeto: **gcc -c**
 - Geração de um código executável a partir de vários códigos-objeto.

5. Alocação dinâmica de endereços de memória

- Endereços de memória: variáveis do programa;
- Alocação estática: endereços gerados pelo compilador
- Alocação dinâmica
 - O que é alocação dinâmica de endereços de memória
 - Por que é necessário? Situações em que é necessário;
 - Função **malloc**
 - O que e faz a função **malloc**;
 - Como a função **malloc** aloca memória

- O valor de retorno da função **malloc**
- Como alterar o tamanho de uma área de memória: função **realloc**
- Definição e uso de variáveis do tipo ponteiro
 - Ponteiro para tipos básicos;
 - Atribuição de valor a ponteiros
 - Operadores de ponteiro: **&** e *****
 - Ponteiro para vetores
 - Relação entre ponteiro e vetor
 - Varredura de um vetor por meio de um ponteiro
 - Apresentação do conteúdo de memória
- Aritmética de ponteiros;
 - Uso de ponteiros para variáveis de tipo básico e complexo.

6. Depuração

Utilização do GDB para detectar erros em programa que usam bibliotecas e variáveis do tipo ponteiro.

BIBLIOGRAFIA BÁSICA

ASCENDIO, A. F.; CAMPOS, E. A. Fundamentos da programação de computadores - algoritmos, C/C++ e java. 2ª. edição. Pearson Prentice Hall: São Paulo, 2007.

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

SCHILDT, H. C. Completo e Total. Editora Pearson-Makron Books. 2011.

BIBLIOGRAFIA COMPLEMENTAR

FARRER, H. Algoritmos estruturados. Vol.3, Guanabara Dois S.A., 1984.

JAMSA, K.A. C library bibliotecas. São Paulo, Mc Graw-Hill, 1988.

MIZRAHI, V. V.. Treinamento em linguagem C. Editora Prentice Hall, 2008.

PUGH, K. Programando em linguagem C. São Paulo, Mc Graw-Hill, 1990.

SALIBA, W. L. C. Técnicas de programação: uma abordagem estruturada. São Paulo: Makron, McGraw-Hill, 1992.

APROVAÇÃO

____/____/____

 Coordenador do Curso de Bacharelado em
 Estatística

____/____/____

 Diretor da Faculdade de Computação