

Estrutura de Dados

ADS

Prof. Dr. Eliseu LS

CONTEÚDO PELO PLANO DE ENSINO:

Alocação dinâmica e ponteiros, Arquivos, Tipos abstratos de dados: conceitos, operações, representações, manipulação, listas, pilhas e filas. Estruturas de representação de grafos. Estruturas para representação de árvores. Árvores binárias e suas aplicações.

LIVROS OBRIGATÓRIOS:

FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico, Lógica de Programação, construção de Algoritmos e Estrutura de Dados

MANZANO, José Augusto N. G., Estudo Dirigido: ALGORITMOS.

MIZRAHI, Victorine Viviane, Treinamento em Linguagem C++ - Módulo 1

Referência: www.cplusplus.com/reference

Notas de aulas da disciplina.

METODOLOGIA DAS AULAS

1. Recursos obrigatórios: Internet, computador e smartphone.
2. **Códigos de programas deverão ser digitados, compilados, executados, impressos digitalmente em um arquivo do ms-word ou PDF, a ser anexado somente pelo Teams, obrigatoriamente com os prints de execução e enunciado do código.**
3. Cada atividade valerá de 2 até 4 pontos e a clonagem de trabalhos de colegas é proibida, será descontado 1,0 de cada tópico não entregue na atividade.

DINÂMICA (Aulas práticas)

1. ***Aula interativa com exercícios de Aprendizagem:*** Nesta parte, o professor apresenta o princípio de funcionamento de cada comando e também os conceitos de lógica, em seguida exibe no datashow, de forma dinâmica ou estática, um exercício de aprendizagem contendo os conceitos de lógica e comandos que foram ensinados.
2. ***Aula dinâmica com exercícios de Fixação:*** Nesta parte da aula, caberá aos alunos a prática dos comandos e conceitos explicados pelo professor. Cabe ao aluno se esforçar para fazer os exercícios e poderá recorrer ao professor que estará disponível para tirar dúvidas via chat para sanar possíveis defeitos no código.
3. ***Oficina de programas/ Avaliação contínua :*** O processo de correção destas atividades se dará de forma individualizada e pessoal pelo professor que apresentará a pontuação e os possíveis comentários de feedback na aula ou pelo ms-teams. **O trabalho é uma avaliação que deverá ser feita preferencialmente no laboratório da faculdade, porém os alunos terão até 4 dias para entregar a atividade caso não consigam terminá-la em aula. Trabalhos atrasados terão automaticamente uma redução de 50% da pontuação, não serão aceitos trabalhos com mais de 2 dias de atraso.**

CORREÇÃO DOS PROGRAMAS/AVALIAÇÕES

a) A nota dos programas/avaliações será considerada nula (Zero), quando:

1. o enunciado não for atendido na íntegra;
2. o código não contiver bibliotecas ou a diretiva namespace;
3. o código não contiver a sub rotina principal `int main()`;
4. Número de sub rotinas (void ou não void) estiver diferente do que a o pedido no enunciado;
5. Técnica de programação (Linear ou Estruturada) estiver diferente do que a pedida no enunciado

b) O código das SUB ROTINAS do tipo void ou não void, das avaliações, será considerado totalmente errado quando:

1. A lista de parâmetros ou argumentos não estiver de acordo com o enunciado;
2. Os parâmetros e argumentos estiverem incorretos ou declarados sem necessidade;
3. contiver fórmulas erradas ou montadas de forma incorreta

COMPOSIÇÃO DA NOTA

Avaliações CP, CC:

1. **PE1:** nota prática, possui peso **0.30**, onde será avaliado em cada tarefa o conhecimento prático do aluno;
2. **PE2:** nota cognitiva, possui peso **0.35**, onde será avaliado dentro de cada tarefa o conhecimento teórico sobre o assunto abordado na tarefa;
3. **TRAB:** Lista de exercícios, cada exercício vale de 0 até 4 pontos, 100% dos pontos será 10. Possui peso de 0.35 na média final.
4. **MEDIA:** $MEDIA = (PE1 * 0.3 + PE2 * 0.4 + TRAB * 0.3) / 2;$

ATENÇÃO: Para a aprovação o aluno deverá alcançar a média 6.

1

Revendo Sub-Rotinas em C/C++

Prof. Dr. Eliseu LS

O que precisa saber antes ?

- 1) Bibliotecas: `iostream`, `iomanip`, `math.h`, `cstdlib` como declarar e para que servem.
- 2) Variáveis e constantes, como declarar e para que servem.
- 3) comandos `cin`, `cout`, `system` para que servem
- 4) Operadores relacionais, lógicos e o comandos `IF` e `GOTO`

O que iremos estudar nesta aula ?

- 1) Declaração de variáveis globais e locais
- 2) Codificação de VOIDS
- 3) Codificação de Functions
- 4) Uso de parâmetros e argumentos
- 5) Quadro Resumo de Sub Rotinas/Declarações de Funções

Escopo e Variáveis Globais

- Um programa C++ é dividido em funções:
 - As definições/declarações no interior funções são **locais**.
 - As definições/declarações fora das funções são **globais**.
- Todo programa precisa ter uma função chamada **main**
 - Esta função é a primeira a ser chamada quando um programa é executado.
 - Sintaxe:

```
// exemplo.cpp  
  
int main(void)  
{  
    Bloco de comandos  
    return(0);  
}
```

← Comentário
← Espaço global
← Função principal
← O símbolo ; indica fim de linha

Todo código fonte para ser executado deve ter a função **main ()**, entretanto você poderá criar suas próprias **sub rotinas** que podem ser functions ou voids, cada uma com um objetivo específico, para **leitura**, **cálculos** ou **exibição**.

As sub rotinas devem ser declaradas logo abaixo das diretivas do pré-processador e acima da função **main ()**, exatamente no **Escopo global** onde também serão criadas as **constantes** e **variáveis globais**, que **permanecerão na memória durante toda a execução do programa**.

Porém o código das sub rotinas deve ficar abaixo da função **main()**, isto é uma boa prática de programação.

Sub rotina do tipo Void e Variáveis Locais

O **tipo void** (vazio) pode ser considerado um pequeno programa dentro de um programa maior, é executado quando o seu nome é digitado em outra sub rotina qualquer, seu código será finalizado pelo comando **return**; **Variáveis locais** são criadas dentro sub rotinas, com escopo local, não podem ser acessadas fora das sub rotinas onde foram criadas. Ocuparão a memória somente durante o tempo de execução das sub rotinas onde foram criadas.

Simulação 1 - Tipo Void

```
double const pi = 3.14; // constante
```

```
double altura = 5.55; // variável global
```

```
void verVolume ( double raio ); // declaração
```

```
int main() {
```

```
double raio = 10.5;
```

```
verVolume ( raio ); // chamada por  
referência
```

```
verVolume ( 13.4 ); // chamada por valor
```

```
system("sleep 5"); }
```

```
// código do void
```

```
void verVolume ( double raio )
```

```
{ double volume = raio * raio * pi * altura;
```

```
cout<< "\nVolume :" << volume << endl;
```

```
return; }
```

Sub Rotina do tipo FUNCTION (não void)

Uma **função** (tipo int, double, string, char, float) têm o objetivo de produzir e armazenar o seu próprio valor através de seu código interno. Este único valor será armazenado através do comando **return** precedido de seu valor de retorno. Para executar/chamar uma function usa-se uma variável local do mesmo tipo da função dentro de uma subrotina qualquer.

Simulação 2 - Function (Não void) "Incluir um Menu"

```
int lern1();
```

```
int modulo (int numero);
```

```
int main ( ) {  
    int num, modu;  
    num    = lern1 ( );  
    modu = modulo ( num );  
    cout << modu << endl; system("sleep 5"); }
```

```
int lern1 ( ) { int n1;  
    cout << "Digite n1: ";  
    cin >> n1; return n1;}
```

```
int modulo ( int numero ) {  
    int modu;  
    if ( numero < 0 ) modu = numero * -1;  
    else modu = numero;  
    return modu; }
```

2 REGRAS PARA MEMORIZAR

1 – Parâmetros, são variáveis locais de ENTRADA de uma sub-rotina qualquer declaradas entre os parênteses.

2 – Entre os parâmetros de uma sub-rotina, não devem haver **constantes**, **variáveis globais** ou **variáveis internas de saída ou resultados** de fórmulas internas.

Recomendo: *Assistam os vídeos Aula 3 Programação Estrutura em C++ partes 1 e 2*

Canal: Eliseu Lemes C++ (PlayList Aulas de C++)

TAREFA / AVALIAÇÃO CONTINUADA

1. Faça o programas h, i, k, l e m da página 26 do livro de exercícios, não necessita menu, somente crie as sub-rotinas e execute a partir da função `int main()`. Não se esqueça de declarar as sub-rotinas antes da função `int main()` e codificar as sub-rotinas após o `int main()`.
2. Faça o código fonte para ler o comprimento e calcular o diâmetro, calcular o raio e finalmente a área de um círculo; (Faça um menu com as opções necessárias. Utilize a simulação 3 como referência/Modelo para fazer este trabalho.)

NOTA: Entregar as atividades juntamente com os enunciados, códigos e prints de execução em um arquivo do tipo DOCX ou PDF, via Teams ou Google Home Class.