



Residência
em Software

Dúvidas e Correções sobre Práticas Anteriores

Professores:

Álvaro Coelho, Edgar Alexander,
Esbel Valero e Hélder Almeida

INSTITUIÇÃO EXECUTORA



UESC

COORDENADORA



APOIO

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Resumo das aulas anteriores

- O código fonte de um programa escrito em **C++** deve ser editado num editor de texto comum ou numa **IDE** apropriada (VS Code por exemplo);
- O arquivo de texto contendo o código fonte é formado, em princípio, por um conjunto de instruções contidas numa única linha e delimitadas por ponto e vírgula (;) utilizadas para:
 - A inclusão dos arquivos de cabeçalho: é opcional e depende de quais recursos serão utilizados na implementação;
 - A declaração de variáveis globais: é opcional e depende da escolha do programador pela melhor estratégia para uso de variáveis no algoritmo;
 - A declaração da função main e de seu código fonte delimitado por { ... } : a declaração da função e de seu bloco sintático pode-se estender por várias linhas e no fechamento do bloco (}) não é necessário colocar um ponto e vírgula (;).

Resumo das aulas anteriores

- A função `main` é o ponto de entrada do programa e o código da mesma é formado por um conjunto de instruções, cada uma delas contidas numa única linha e delimitadas por ponto e vírgula (;)
- As variáveis a serem utilizadas na implementação tem que ser declaradas antes de serem utilizadas;

- `tipo nome_da_var1;`
- `tipo nome_da_var2, nome_da_var5, nome_da_var4;`
- `tipo nome_da_var5 = valor_inicial;`

Resumo das aulas anteriores

- O tipo de uma variável em **C++** pode ser, em princípio, um dos tipos básicos:
 - Tipo caractere: `char` ou alguma das suas variações;
 - Tipo inteiro: `int` ou alguma das suas variações;
 - Tipo ponto flutuante de dupla precisão: `double` ou alguma das suas variações;
 - Tipo booleano: `bool`;
 - Tipo vazio: `void`;

Resumo das aulas anteriores

- Uma variável pode ser inicializada durante sua declaração ou o valor nela contido pode ser alterado utilizando o operador atribuição (=);
 - `identificador_de_variavel = expressão;`
 - `var1 = var2 = var3 = var4 = expressão;`
 - em um comando de atribuição o valor do **rvalue** é convertido no tipo do **lvalue**, antes de fazer a atribuição;

Resumo das aulas anteriores

- As expressões em **C++** podem combinar os valores de diferentes variáveis ou literais utilizando operadores válidos da linguagem:
 - Aritméticos;
 - Relacionais;
 - Lógicos;
 - Bit a Bit
- Podemos utilizar ainda o operador condicional (?);

Atividades

- Os caracteres numéricos aparecem na tabela **ASCII**, e em outras, numa sequência que começa pelo caractere `'0'` até o caractere `'9'`. As variáveis de tipo caractere podem ser tratadas também como valores numéricos. Com base nestas afirmações desenvolva um programa em C++ que:
 - Imprima na tela, utilizando `cout`, cada um dos caracteres numéricos e seu correspondente código numérico. Como modificar o comportamento do `cout` para imprimir um objeto de tipo `char` como caractere e como número?

Exemplo:

<code>'0'</code>	-	48
<code>'1'</code>	-	49
...		
<code>'9'</code>	-	57

Atividades

- b. Modifique o exercício anterior para que a saída imprima também o código numérico em octal e em hexadecimal.
- c. Acrescente ao código do exercício anterior a possibilidade de ler um caractere qualquer e imprima no mesmo formato do inciso anterior.
- d. Pesquise como fazer para armazenar uma variável os caracteres especiais `'ç'` e `'ã'`. Acrescente no código do exercício anterior um exemplo que demonstra como usar este recurso.

Atividades

- Durante a aula foi apresentado o tipo de dados `int` e suas variações, que permitem representar um subconjunto dos números inteiros. Sobre estes tipos de dados crie uma aplicação que:
 - a. Sobre o tipo `int` identifique qual o menor e o maior valor que pode ser representado por uma variável deste tipo. Mostre esta informação na tela.
 - b. Crie uma variável `uli`, de tipo `unsigned long int`, atribua a ela o maior valor que pode ser armazenado neste tipo de dado (identifique qual o menor e o maior valor que pode ser representado com um `unsigned long int`);

Atividades

- b. Crie uma variável `li`, de tipo `long int`, atribua a ela o valor da variável `uli` do exercício anterior, e depois atribua a `uli` o valor armazenado em `li`. O que acontece e por que? Mostre o resultado na tela e coloque suas considerações num comentário no código.
- c. Atribua a `li` o maior valor que pode ser armazenado num `long int` (identifique qual o menor e o maior valor que pode ser representado com um `long int`);

Este tipo de análise pode ser estendido aos tipos `unsigned long int` (como está na continuação do exercício) ou ao `short int` e `unsigned short int`

Discussão sobre o tema ...

Atividades

- Durante a aula foram apresentados o tipo de dados `double` e suas variações, que permitem representar um subconjunto dos números reais. Sobre estes tipos de dados crie uma aplicação que:
 - a. Sobre o tipo `float` identifique qual o menor e o maior valor que pode ser representado por uma variável deste tipo. Mostre esta informação na tela.
 - b. Crie uma variável `pi_f` de tipo `float` e atribua a ela o valor da constante **pi** com o maior número de casas decimais que você consiga achar na Internet. Imprima este valor na tela.

Atividades

- c. Explore as formas de formatar a saída com o comando `cout` e modifique a quantidade de casas decimais que são apresentadas, na impressão do valor de **pi**, para 2, 4, 8 e 16.
- d. Crie uma variável `pid` de tipo `double` e atribua a ela o valor da constante **pi** com o maior número de casas decimais que você consiga achar na Internet. Imprima este valor na tela.
- e. Identifique se existe alguma diferença entre o valor `pif` e `pid` quando seus valores são impressos com `cout` utilizando 2, 4, 8 e 16 casas decimais
- f. No seu sistema existe diferença entre o tipo `double` e `long double`? Mostre seu achado com algum exemplo dentro do seu código.

Atividade

- Escreva um programa que:
 - a. Declare três variáveis de tipo `char`, vamos chamar de `ch1`, `ch2` e `ch3`, mas você pode escolher outro identificador se achar mais apropriado;
 - b. Peça ao usuário para digitar um caractere, obtenha o mesmo da entrada padrão utilizando `cin`, e atribua o valor digitado à variável `ch1`;
 - c. Verifique, utilizando o operador condicional (`?`) se se trata de:
 - i. uma letra maiúscula;
 - ii. uma letra minúscula;
 - iii. um dígito;
 - iv. outro tipo de caractere;

Atividades

- d. Atribua à variável `ch2` o caractere `81`, identifique e imprima na tela, utilizando `cout`, o caractere em formato numérico decimal, octal, hexadecimal e como caractere;
- e. Dado que o caractere `ch2` é uma letra maiúscula, atribua à variável `ch3` o caractere que corresponde à mesma letra minúscula (não pode procurar na tabela ASCII) e Imprima `ch3` na tela, utilizando `cout`, em formato numérico decimal, octal, hexadecimal e como caractere;

Atividades

- Escreva um programa, usando quando necessário o operador (?), que:
 - a. Declare três variáveis de tipo `int`, vamos chamar de `a`, `b` e `c` mas você pode escolher outro identificador se achar mais apropriado;
 - b. Peça ao usuário para digitar dois números inteiros, obtenha-os da entrada padrão, usando `cin`, e atribua os valores digitados às variáveis `a` e `b` respectivamente;
 - c. Atribua à variável `c` a soma das variáveis `a` e `b`, imprima o resultado na tela em formato hexadecimal;
 - d. Atribua à variável `c` o produto das variáveis `a` e `b`, imprima o resultado na tela em formato octal;

Atividades

- e. Atribua à variável `c` o módulo (valor absoluto) da diferença entre as variáveis `a` e `b`, imprima o resultado na tela; (valor absoluto: se `c < 0` retorna `-c`, caso contrário retorna `c`)
- f. Atividade avançada: Verifique se o **C++** fornece algum tipo de recurso na sua biblioteca padrão para obter este tipo de resposta de forma direta;
- g. Atribua à variável `c` o quociente entre variáveis `a` e `b`, imprima o resultado na tela; (se `b` for igual a zero a divisão não é possível e um aviso deve ser apresentado ao usuário);
- h. Se a divisão anterior for possível, determinar se `a` é divisível de forma exata por `b`, mostrar o resultado na tela;

Atividades

- Escreva um programa que:
 - a. Peça ao usuário para fornecer um número inteiro, obtenha-o da entrada padrão, usando `cin`, e armazene numa variável de tipo `int`;
 - b. Determine se o valor fornecido pode ser representado, sem perda de informação, como um `short int`. Imprima na tela o valor fornecido, seguido das palavras “e maior que um `short int`” ou “este valor pertence ao intervalo dos `short int`”. Use apenas o operador condicional (`?`).

Atividades

- Escreva um programa que:
 - a. Declare três variáveis de tipo `double`, vamos chamar de `x`, `y` e `z` mas você pode escolher outro identificador se achar mais apropriado;
 - b. Peça ao usuário para digitar dois números de ponto flutuante, obtenha-os da entrada padrão, usando `cin`, e atribua os valores digitados às variáveis `x` e `y` respectivamente;
 - c. Considerando que `x` e `y` são coordenadas num plano cartesiano, identifique em que lado da curva $f(x) = 5x + 2$ se encontra (esquerda, direita ou na curva). Imprima o resultado na tela;

Atividades

- d. Atividade avançada: Atribua à variável z o valor da distância euclidiana do ponto (x, y) ao centro de coordenadas; (pesquise na biblioteca padrão como determinar a raiz quadrada de um número)
- f. Atribua à variável z o produto entre as variáveis x e y , imprima o resultado na tela em notação científica;

Atividades

- Uma pesquisa monitora 8 genes de uma planta para avaliar como cada um deles influencia na resposta a uma determinada praga. Os pesquisadores optaram por representar a informação dos genes de cada planta com um inteiro sem sinal, onde cada bit da representação binária identifica se um gene está presente (1) ou não (0) naquela planta. Implemente um programa que:
 - a. Leia a informação genética de uma planta na forma de um caractere sem sinal (`unsigned char`);
 - b. Utilizando os operadores bit a bit identifique quantos, dos genes estudados, estão presentes;
 - a. Solicite do usuário a informação sobre qual gene específico ele quer informação (um valor inteiro entre 1 e 8) e retorne se este gene se encontra nesta planta.