UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PET - PROGRAMA DE EDUCAÇÃO TUTORIAL - ENG. ELÉTRICA

CIT 2015.2

Aula 02 – Variáveis e Estruturas de desvio condicional

Autor: Max Rodrigues Marques

Carga Horária: 2 h

Variáveis

- Uma variável nada mais é que um nome que damos a uma determinada posição de memória para conter um valor de um determinado tipo.
- Variáveis possuem tipos, os quais determinam o espaço que as variáveis irão ocupar na memória.
- Uma variável deve ser sempre definida antes de ser usada. A definição de variáveis é feita assim:

```
tipo var,[var2,var3,..,varn];
```

Tipos de dados no C++

- C++ tem 6 tipos de dados primitivos, a saber: *char*, *int*, *float*, *double*, *wchar_t* e *bool*.

- A partir desses tipos de dados, podemos formar tipos de dados compostos: vetores, matrizes, *strings* e structs.

- Veremos a seguir mais detalhes sobre cada tipo.

Tipo int

- É relacionado com os números inteiros.
- A declaração e feita através da palavra reservada “int”.
- O tipo inteiro pode armazenar números que vão da faixa de -32768 a +32767.

```
int inteiro = 500;
```

Tipo float (precisão simples)

- Representa números fracionários e números reais, (o que inclui os números inteiros).
- A declaração é feita pela palavra reservada float.

```
float real = 1.15479637; // poucas casas decimais.
```

Tipo double (dupla precisão)

- Igualmente ao tipo *float*, utilizado para números fracionários e reais, porém com precisão ainda maior.
- A declaração é com a palavra reservada *double*:

```
double numero = 558749.16846516487975132156857452131274127412974812794812794812798412;
// precisão acurada.
```

Tipo lógico (booleano) bool

- É o tipo mais simples, e pode armazenar apenas dois valores: *verdadeiro (true)* ou *falso (false)*.

```
bool logico1 = true; // True significa verdadeiro em inglês;
bool logico2 = false; // False significa falso em inglês;
bool logico3 = 0; // Atribui-se o inteiro 0 (zero) equivale a 'false';
bool logico4 = 189; // Qualquer valor diferente de 0 (zero) é interpretado como 'true';
```

Tipo char (caracter ASCII)

•O tipo de dados caracter é utilizado para armazenar letras, números e outros caracteres especiais.

• A declaração do tipo dá-se através da palavra reservada *char*.

•Ao total, são 256 caracteres que podem ser armazenados, dos quais os 128 primeiros fazem parte do conjunto ASC II.

```
char umCaracter = 'a';
```

Modificadores de tipo

• Você pode modificar a faixa de valores usando os seguintes modificadores de tipo: *signed*, *unsigned*, *short* e *long*.

•short: fixa a faixa dos valores inteiros (*int*) para -32.768 até 32.767.

•long: Aumenta a faixa de valores. Um tipo int já é um long int, um long float será um double.

•signed e unsigned: Estes modificadores alteram a faixa dos números inteiros (*int*) e caracteres (*char*) para valores somente positivos (*unsigned*) ou fixa a faixa deles em valores com sinal (*signed*).

```
unsigned int numero1 = 700; // inteiro sem sinal.
signed int numero2 = -700; // admite valores negativos (padrão do tipo int).
unsigned short int numero3 = 5000; // pode-se combinar modificadores.
signed short int numero4 = 22000;
signed long int numero5 = 454658;
unsigned long int numero6 = 3000000000; // aceita somente inteiros positivos.
signed char caracter1 = -128; // caracter com sinal. Faixa: -128 até 127.
unsigned char caracter2 = 255; // caracter sem sinal. Faixa: 0 até 255.
```

Operador de atribuição

•Quando uma variável é declarada fica sempre com um valor, o qual resulta do estado aleatório dos bits que a constituem. Além do mais, uma variável poderá ser iniciada com um valor através de uma operação de atribuição.

• Em C++, o operador de atribuição é o "=", onde a atribuição é realizada obedecendo a seguinte sintaxe:

tipo variável = expressão

Operadores aritméticos

•Usados, como o próprio nome sugere, para realização de operações matemáticas.

Adição	A+B
Incremento pré- fixado	++a
Subtração	A-B
Decrementação pré-fixada	--a
Multiplicação	a*b
Divisão	a/b
Resto	a%b

Operadores comparativos

- Estabelece comparação entre variáveis.
- Usado corriqueiramente em estruturas de desvio condicional.

Operador	Sintaxe
Menor que	`a < b`
Menor ou igual que	`a <= b`
Maior que	`a > b`
Maior ou igual que	`a >= b`
Diferente de	`a != b`
Igual a	`a == b`
Não lógico	`!a`
E lógico	`a && b`
Ou lógico	`a || b`

Estruturas de desvio condicional

• O desvio condicional é uma estrutura de fluxo onde o programa analisa uma condição e:

-caso esta condição seja verdadeira executa alguns comandos.

-caso esta condição seja falsa executa outros comandos.

Estrutura de desvio *if-else*

- A instrução if-else é uma das instruções de controle de fluxo.
- Permite indicar quais circunstâncias em que se deve executar instruções ou conjuntos delas.
- A sintaxe é:

```
if (condição)
 {
   instrução 1...;
 }
else instrução 2...;
```

Ex:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int var=10;
8      if(var==10)
9      {
10      cout<<"O valor da variavel eh:"<<var<<endl;
11     }
12     else cout<<"O valor da variavel nao eh"<<10<<endl;
13    return 0;
14 }
15
```

Instruções if-else encadeadas

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int salario=10;
8      cout<<"Qual eh o valor do salario?"<<endl;
9      cin>>salario;
10     if(salario<0)
11     {
12      cout<<"Valor não valido"<<endl;
13     }
14     else if(salario>2000)
15     {
16       cout<<"Imposto "<<salario*0.10<<endl;
17     }
18     else cout<<"Imposto "<<salario*0.05<<endl;
19    return 0;
20 }
```

Prática

1- Escreva um programa que leia dois números e os apresente por ordem crescente.
2- Escreva um programa que recebe um inteiro e diga se ele é par ou ímpar.
3- Escrever um programa que recebe três notas individuais em três aspectos de um estudante, calcula a média final ponderada com pesos 2, 3 e 4. O programa deve exibir um relatório com o nome do aluno, suas notas individuais, sua média final e dizer se o aluno está aprovado ou não. A média para aprovação é 7.

Estrutura condicional switch

- A instrução switch adapta-se à tomada de decisões em que o número de possibilidades é elevado.
- Sua sintaxe é:

```
switch (expressão)
{
    case constante1: instruções1;
    case constante2: instruções2;
    ..................
    case constanten: instruçõesn;
    default: instruções;
}
```

Exemplo switch:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7     int idade=0;
8     cout << "Qual eh a sua idade? " << endl;
9     cin>>idade;
10    switch(idade)
11    {
12       case 17: cout<<"Voce tem 17, que legal..."<<endl;
13       case 18: cout<<"Voce tem 18, isso é bom, creio "<<endl;
14       case 19: cout<<"Tá quase na mesma dos 18..."<<endl;
15       default: cout<<"tá certo, acredito!"<<endl;
16    }
17
18    return 0;
19 }
```

• Ao usar o switch como apresentado acima, o programa se comporta de um modo estranho. Isso se deve ao fato da falta de um outro elemento, o break.

break

•A instrução break permite parar a execução dentro do **switch**, continuando o programa na instrução seguinte ao **switch**.

• A sintaxe com o break é:

```
switch (expressão)
{
        case constante1: instruções1; break;
        case constante2: instruções2; break;
        ..................
        case constanten: instruçõesn; break;
        default: instruções;
}
```

Exemplo usando switch com break

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7     int idade=0;
8     cout << "Qual eh a sua idade? " << endl;
9     cin>>idade;
10    switch(idade)
11    {
12       case 17: cout<<"Voce tem 17, que legal..."<<endl; break;
13       case 18: cout<<"Voce tem 18, isso é bom, creio "<<endl; break;
14       case 19: cout<<"Tá quase na mesma dos 18..."<<endl; break;
15       default: cout<<"tá certo, acredito!"<<endl; break;
16    }
17
18    return 0;
19 }
```

Prática

1- Escreva um programa que dada uma letra, escreva na tela se essa letra é ou não uma vogal. Dica: a função toupper(letra) converte um char para caixa alta (maiúsculo) e a função tolower(letra) converte um char para caixa baixa (minúsculo).

2- Escreva um programa em que o usuário entra com o valor da dezena de sua idade e programa imprime as seguintes mensagens de acordo com o valor digitado:

se dezena=0, imprime “Criança”

se dezena=10, imprime “Adolescente”

se dezena=20, imprime “Jovem”

se dezena=30, imprime “Adulto”

se dezena=60, imprime “Melhor idade”

Acima desse valor, imprima “Aposentado”

Dúvidas?