

# UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

## Campus CAMPO MOURÃO

### Resenha Capítulo 2 - APOO

#### Descrição de recursos de C++ não relacionados às classes

Borges e Clinio

Estudante: Reginaldo Gregório de Souza Neto

RA: 2252813

Mudanças da linguagem C para C++:

Comentários podem ser feitos através de duas barras.	<i>//comentando aqui</i>
Variáveis podem ser declaradas a qualquer momento.	<i>for (int i=0; i&lt;20; i++){}</i>
Não é mais necessária a utilização do typedef.	<i>struct a{... };</i>
União anônima de funções.	<i>Union {int a, chat* str};</i>
Uma função só pode ser usada se já foi declarada.	<i>void f(int a, float b);</i>
Funções que não recebem parâmetros.	<i>float f(); É IGUAL float f(void);</i>

Funções *inline* são funções normalmente pequenas, pois evita geração de código para a chamada e o retorno. Tendo em vista que o compilador inclui junto ao código de execução do programa, o que está escrito dentro da função. No caso da *inline*, a chamada da função é substituída pelo corpo da mesma. Mas isso só pode acontecer caso a função seja implementada no *header* do código, pois o compilador precisa conhecer o corpo da função antes de implementá-lo no lugar da chamada. Se o compilador julgar que a função é muito grande ou complexa, ela é tratada como uma função normal.

Referências se assemelham à ponteiros, pois possuem a característica de “referenciar” algum objeto, tornando capaz a modificação de seu conteúdo através de outro “nome” que não fosse o seu criado originalmente. Para uma referência funcionar ela não deve ser atribuída a um valor constante. Elas podem ser utilizadas em variáveis locais, serem passadas por parâmetro ou então em retornos de funções.

Alocação de memória condiz com a capacidade de armazenamento de memória realizado pelo código dinamicamente, ou seja, o algoritmo “reserva” um espaço na memória para garantir que ele exista e possa ser utilizado durante a execução do programa. Substituição do *malloc* e *free* pelas palavras reservadas *new* e *delete*.

<i>int * i1 = (int*)malloc(sizeof(int));</i>	<i>// C</i>
<i>int * i2 = new int;</i>	<i>// C++</i>
<i>free(i1);</i>	<i>// alocado com malloc (C)</i>
<i>delete i2;</i>	<i>// alocado com new (C++)</i>

Valores default para parâmetros de funções se trata da definição de parâmetros pré-determinados pelo programador para a função, ou seja, caso ela não receba a quantidade de variáveis suficientes por parâmetro, ela utilizará esses valores pré-definidos para ser executada. A declaração do valor default só pode aparecer uma vez. Isto deve ser levado em consideração ao definir funções com protótipos.

Sobrecarga de nomes de funções é um recurso que permite uma função ter mais de um significado, ou seja, o “nome” de uma função pode executar várias implementações diferentes. Para isso, é necessário que haja a implementação de todas essas funcionalidades no programa.

Parâmetros de funções não utilizados se tratam da não utilização dos parâmetros para a computação dentro das funções. Para evitar os avisos do compilador, basta omitir o nome do parâmetro que não será utilizado.

O operador de escopo é uma ferramenta muito útil para possibilitar o acesso a variáveis declaradas em outros escopos, ele permite “atravessar” essas barreiras e acessar o que está dentro de “outra vizinhança”. Sua semântica é: *Escopo desejado :: nome da variável*.

#### REFERÊNCIA:

Borges, Renato, and André Luiz Clinio. "Programação Orientada a Objetos com C+." *Apostila, Rio de Janeiro*.