> Há em todos um indivíduo que ascende por seus próprios esforços de sua posição anterior na vida para uma maior.
>
> Adolf Hitler

1 Escrevendo o seu primeiro código em *C++*

Todo programa em C++ possue alguma características importantes (nem todas obrigatórias), pegando como exemplo o programa 1.

Programa 1: Primeiro programa em C++

```
1  // Game over
   // Meu primeiro programa em C++ (virão outros)
   // programa_001.cpp

   #include <iostream>
6
   using namespace std;

   int main()
   {
11   cout << "Game Over!" << endl;
     return 0;
   }
```

Neste programa é possível perceber algumas coisas:

- Existe alguns comentários;
- Existe um certo padrão de digitação;
- O programa está estruturado (indentado);

1.1 Comentado o seu código

Como pode ser observado no programa 2, as primeiras três linhas do programa são comentários:

Programa 2: Primeiro programa em C++ (comentários)

```
1  // Game over
2  // Meu primeiro programa em C++ (virão outros)
3  // programa_001.cpp
```

O uso dos caracteres // indica que a partir daquele ponto existe um comentário. Qualquer caracterer, símbolo ou comando será tratado como um comentário, e portanto, o compilador irá ignorar o restante da linha.

Também é possível utilizar os caracteres $/*$ e $*/$ para colocar comentários em várias linhas (programa 3). Quaisquer textos colocados entre estes dois símbolos serão ignorados pelo compilador.

Programa 3: Primeiro programa em C++ (comentários)

```
/* Game over
   Meu primeiro programa em C++ (virão outros)
   programa_001.cpp */
```

1.2 Estruturando o programa (utilizando espaços em branco)

A indentação (distância das margens) são espaços que um programador deixa no início de cada bloco de programa. A indentação não é requerida pela linguagem C++. Entretanto, seria ótimo que todos programadores utilizassem (3 espaços são o suficiente), observe os programas 1 e 4:

Programa 4: Primeiro programa em C++ (sem identação)

```
// Game over
// Meu primeiro programa em C++ (virão outros)
// programa_004.cpp


#include <iostream>

using namespace std;

int main()
{
cout << "Game Over!" << endl;
return 0;
}
```

O uso da indentação (uso de espaços em branco) não afetará o desempenho e funcionalidade do seu código. Mas com certeza aumetará a legibilidade do seu programa.

1.3 Incluindo outros arquivos

A próxima linha do programa (programa 5) é uma diretiva do pré-processador (pré-compilação). Na fase de pré-compilação, o programa fonte é lido e caso se encontre comandos do pré-compilador, eles serão processados. O pré-compilador gera então um código intermediário que será lido pelo compilador.

Programa 5: Primeiro programa em C++ (pré-processador)

```
#include <iostream>

using namespace std;
```

Todos os comandos para o pré-compilador começam com o caractere ♯ (jogo da velha ou tralha). O comando ♯*include* indica para o pré-compilador ler o arquivo indicado e colocar o mesmo no programa fonte intermediário.

O arquivo incluído possui o nome de arquivo *header* (cabeçalho) e geralmente possui protótipo (assinatura) de funções a serem utilizadas por todos os programas de um sistema. Possui também as declarações de tipos existentes (*typedef*) no sistema.

1.4 Definindo a função *main()*

A próxima linha linha especifica o uso de uma função chamada *main()* (programa 6). Todo programa em C++ deve ter uma função chamada *main*. É por esta função que será iniciada a execução do programa.

Programa 6: Primeiro programa em C++ (função *main*)

```
1 int main()
{
  cout << "Game Over!" << endl;
  return 0;
}
```

Uma *função* é um grupo de código que cumpre uma determinada atividade e retorna um valor. Deve-se especificar o tipo de retorno da função, que pode ser *int* ou *void* (não se preocupe com estes tipos no momento). Caso seja colocado *int*, o valor retornado pela função main estará disponível teste no sistema operacional. Caso o retorno da função seja declarado como *void*, nada será retornado ao sistema operacional. Alguns compiladores podem exigir que o retorno da função *main* seja declarado como *int*.

Neste caso, o valor *int* indica que a função retornará um valor inteiro (para o sistema operacional).

As linhas entre { e } delimita o início e fim do programa (ou bloco de um programa). Um bloco de programa deve ser indentado para melhor entendimento. No caso de uma função (qualquer função, mas neste caso a função *main*) um trecho entre { e } é chamado de *corpo* da função (programa).

1.5 Mostrando um texto na saída padrão

A próxima linha no corpo da função *main* mostra o texto *Gamer Over!* na tela (saída padrão). *"Game Over!"* é uma *string* (conjunto de caracteres - letras, números ou símbolos - que podem ser impressos). Normalmente está representando entre aspas.

O comando (objeto) *cout*, definido no arquivo *iostream*, que realiza a saída de dados para a saída padrão do sistema operacional. Uma vantagem de uso do objeto é que ele distingue o tipo do dado ou variável a ser mostrada e formata a saída de acordo com este formato.

A indicação dos valores que devem ser impressos via comando *cout* precisam utilizar o operador de saída $<<$. Cada valor, seja constante ou variável, deve ser precedido por este operador. Pode-se colocar mais de um operador de saída para um único *cout*. As informações serão mostradas em seqüência, sem a colocação de nenhum separador entre as saídas.

O manipulador *endl* indica a quebra de linha (mudança de linha).

Para entender a relação entre o arquivo *iostream* e o comando *cout* observe a figura 1:

A linguagem C++ define uma série de comandos (objetos). Estes comandos estão distribuídos entre vários arquivos (para facilitar a documentação e uso). O comando *cout* é um exemplo desta organização. Ele está definido dentro do arquivo *iostream* (*io* $\Longrightarrow$ entrada/saída e *stream* $\Longrightarrow$ manipulação de caracteres e outras informações). O operador $<<$ e o manipulador *endl* também estão definidos dentro do arquivo *iostream*.

Você deve lembrar-se da linha *using namespace std* (usando o espaço de nomes std). Esta linha indica que o programa deve utilizar (abrir) os comandos (objetos) definidos em *std* dentro do *iostream*. Não se preocupe com a terminologia *objetos* neste momento, no momento adequado você aprenderá o que significa *objeto* (lembre-se que C++ é uma linguagem de programação orientada a objetos).

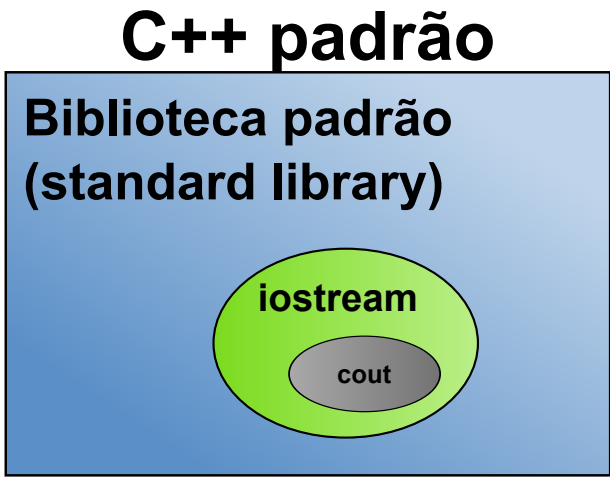


Figura 1: Relação entre *iostream* e *cout*.

1.6 Retornando um valor

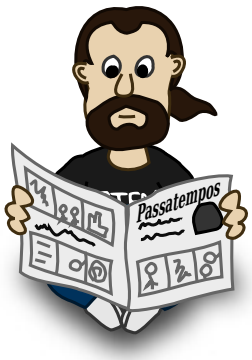
A última linha da função *main* (programa 6) indica que o valor 0 (zero) será retornado ao sistema operacional. Retornar o valor zero indica que o programa terminou sem erros ou problemas.

Erros comuns que pode ser cometidos neste momento:

- Você esqueceu de colocar um ; (ponto e vírgula) ao final da linha de comando;
- Você digitou algum comando em caixa alta (maíscula);

E um detalhe, para você conseguir imprimir uma \ (barra), você deverá utilizar \\ (duas barras seguidas)

2 Exercícios



1. Implemente um programa que escreve na tela a frase 'O primeiro programa a gente nunca esquece!'.
2. Elabore um programa que escreve seu nome completo na primeira linha, o nome da disciplina na segunda linha e o nome da instituição (onde você estuda) na terceira linha.
3. Faça um programa que mostre na tela algumas frases assassinas, que são aquelas que fazem com muitas idéias sejam perdidas antes que amadureçam ou seja aprofundadas. Eis alguns exemplos (bole também os seus):
   - 'Isto não vai dar certo.'
   - 'Você nunca vai conseguir.'
   - 'Você vai se estrepar.'
   - 'Não vai dar em nada.'
   - 'Está tudo errado!'
4. Escreva uma mensagem para uma pessoa de que goste. Implemente um programa que imprima essa mensagem, chame a pessoa e execute o programa na frente dela.
5. Escreva um bilhete ao seu professor, informando seus objetivos nesta disciplina e o que espera dela e do professor. Implemente um programa que mostra seu bilhete na tela.
6. Faça um programa para imprimir um desenho na tela (veja o exemplo):

```
   +---+
  /     \
 /  o o  \
 \   ^   /
  \ === /
   \   /
    +-+
```

7. Elabore um programa para produzir na tela a letra X usando a própria. Se fosse 'L', seria assim:

```
L
L
L
LLLL
```

8. *Emoticons* são seqüências de caracteres que mostram rostos e expressões, vistos de lado e usados freqüentemente em correios eletrônicos e bate-papos na Internet. Existem dezenas; veja alguns:
   - sorriso :-)
   - tristeza :-(
   - mostrando a língua :-p
   - espanto :-o
   - cabelo partido ao meio {:-)
   - usa bigode :-{
   - beijo :-*

   Elabore um programa que mostre na tela os emoticons, um em cada linha, com a descrição de cada um.