

# Capítulos 1 & 2

## Programação e Programas

Bjarne Stroustrup

[www.stroustrup.com/Programming](http://www.stroustrup.com/Programming)

# Sumário

Hoje, apresentaremos os objetivos da cadeira e o programa. Introduziremos a noção básica de programação e daremos exemplos de áreas nas quais o software é crítico para a nossa civilização. Por fim, apresentaremos o programa C ++ mais simples possível e descreveremos como ele pode ser transformado em código executável.

# Resumo

- Objetivos e descrição do curso
- Programação
- "Hello, world!"
- Compilação

# Uma cadeira...

- Sobre programação
- Para principiantes...
- Que querem escrever programas para resolver problemas
  - i.e., pessoas que criam sistemas que possam ser usados por outros
- Que se assume são inteligentes
  - Embora não necessariamente gênios
- Que estão dispostos a trabalhar duramente
  - Embora o peso da cadeira seja normal
- Usando a linguagem de programação C++

# Objetivos

- Ensinar/aprender
  - Conceitos fundamentais de programação
  - Técnicas essenciais
  - Conceitos básicos de C++
- No fim do ano serão capazes de...
  - Escrever programas de complexidade média em C++
  - Ler programas muito maiores
  - Aprender sozinhos as bases de muitas outras linguagens
  - Estudar conceitos avançados de C++
- Não serão...
  - Peritos em programação
  - Peritos em C++
  - Utilizadores especialistas em bibliotecas avançadas.

# Os meios...

- Aulas teóricas, práticas e tutorias
  - Venham às aulas!
- Notas/Capítulos
  - Leiam os capítulos antes da aula
  - Leiam novamente depois das aulas
  - O autor agradece “feedback” (erros tipográficos, sugestões, etc.)

# Os meios (Cont.)

- Trabalho
  - Revejam as perguntas nos capítulos
  - Revejam as definições (“terms”) nos Capítulos
  - Treinos (“drills”)
    - Resolvam sempre os exercícios de treino
    - Façam sempre os treinos antes dos exercícios
  - Exercícios

# Cooperem na aprendizagem

- Não estudem sozinhos (sempre que puderem)
  - Criem grupos de estudo
  - Ajudem-se uns aos outros
- Vão às tutorias e esclareçam dúvidas comigo
  - Levem questões preparadas
  - Não existem perguntas estúpidas!

# Porquê C++ ?

- Não se aprende a programar sem uma linguagem de programação
- O objectivo de uma linguagem de programação é permitir expressar as vossas ideias em código
- C++ é a linguagem que mais diretamente vos permite expressar ideias no maior número de áreas de aplicação.
- C++ é a linguagem mais usada em engenharia (e matemática)
  - <http://www.stoustrup.com/applications.html>

# Porquê C++ ?

- O C++ é definido de forma precisa e abrangente por um padrão ISO.
  - E o standard é quase universalmente aceite
  - A norma mais recente é de 2017 (o C++ 2020 está em estudo).
- O C++ está disponível em quase todo o tipo de computadores.
- Os conceitos de programação aprendidos em C++ podem ser usados, quase diretamente, noutras linguagens
  - Incluindo C, Java, C#, e (menos diretamente) Fortran

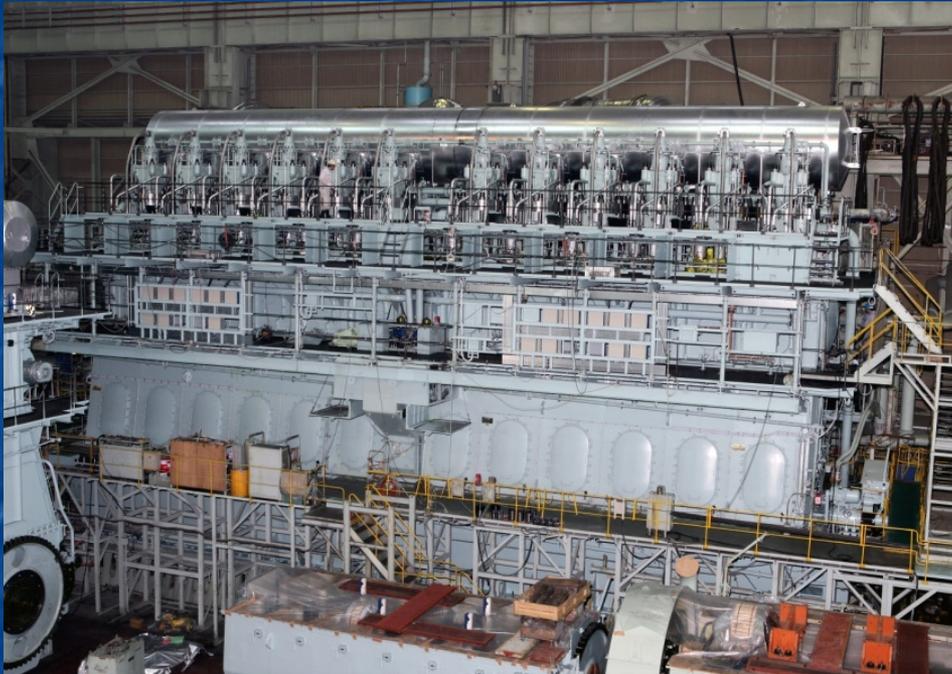
# Programa

- Parte I: Conceitos básicos
  - Tipos, variáveis, “strings”, I/O, cálculos, erros, vetores funções, classes
- Parte II: Input e Output
  - Ficheiros, I/O “streams”
  - Output gráfico (não damos)
  - GUI (Graphical User Interface) (não damos)
- Parte III: Estruturas de dados e algoritmos
  - Memória dinâmica, ponteiros, e “arrays”
  - lists, maps, ordenação e pesquisa, vectors, templates (damos parcialmente)
  - A STL (Standard Template Library)
- Parte IV: Alargando horizontes
  - Software: ideais e história
  - Porocessamento de texto, programação de sistemas embebidos, testes, C, etc. (não damos)

# Porquê programar?

- A nossa civilização depende do software
  - A matemática aplicada depende fortemenete da utilização de software.
- Nota: A maioria dos programas não correm em máquinas que parecem um PC
  - um monitor, um telemóvel, um teclado, uma máquina de lavar, etc.

# Navios



- Desenho
- Construção
- Gestão

- Monitorização
- Motor
- Desenho do casco
- Bombas

# Aviação



- Comunicações
- Controlo
- Visualização
- Processamento de sinais
- Controlo de equipamentos
- Monitorização

# Telefones



- Qualidade do som
- Interfaces
- Faturação
- Mobilidade
- Comutação
- Confiabilidade
- Aprovisionamento
- Imagens

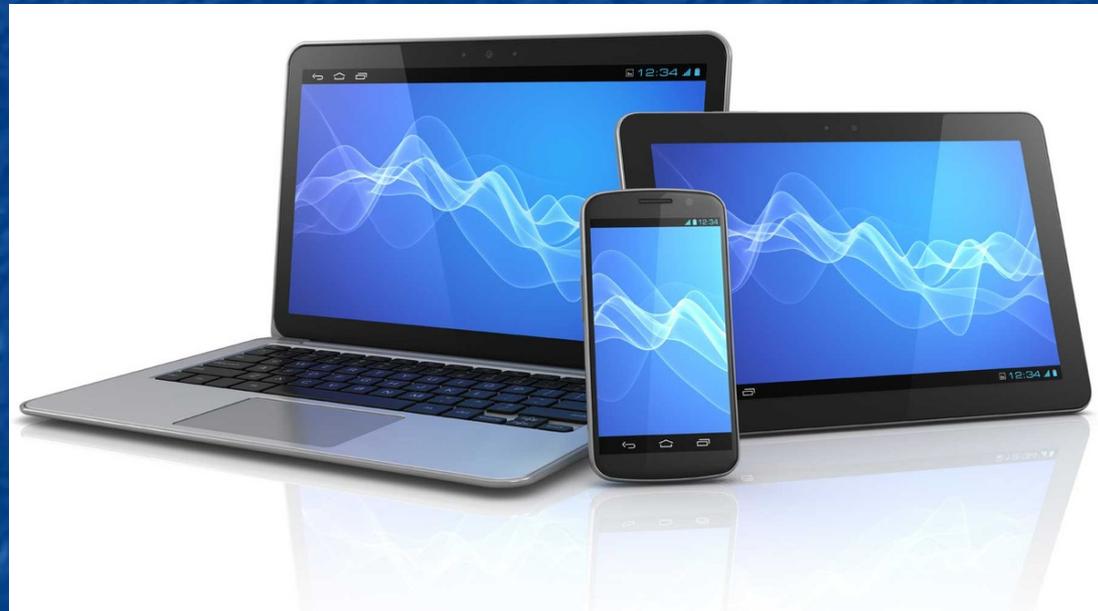
# Energia



- Controlo
- Monitorização
- Análise
- Desenho

- Comunicações
- Visualização
- Fabricação

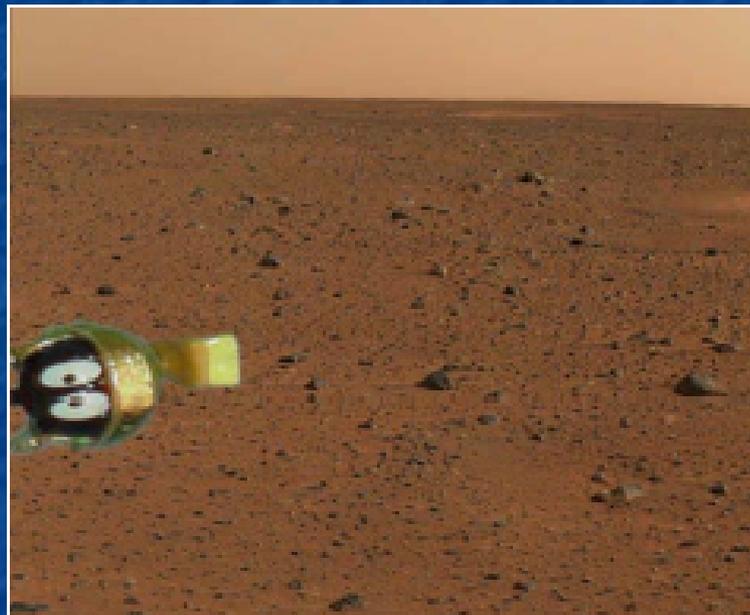
# PC/tablet/workstation



- Há muito mais na computação do que jogos, processamento de texto, “internet browsing” e folhas de cálculo!

# Onde é usado o C++?

- Praticamente em todo o lado



“Mars rovers”, animação, gráficos, Photoshop, GUI, SO, compiladores, slides, desenho e fabrico de processadores, etc.

Ver [www.stoustrup.com/applications.html](http://www.stoustrup.com/applications.html)

# Primeiro programa – só o esqueleto...

```
// ...
```

```
int main()           // main() é onde começa um programa em C  
{  
    cout << "Hello, world!\n"; // envia Hello, world! Para o monitor  
                               // seguido por uma mudança de linha  
    return 0;         // devolve um valor indicando sucesso  
}
```

*// aspas delimitam uma string (cadeia de caracteres)*

*// **NOTA:** aspas “inteligentes” “ ” cusam erros de compilação.*

*// garantam que usam aspas normais " "*

*// \n é a notação para mudança de linha*

# Primeiro programa – completo

*// primeiro programa:*

```
#include "std_lib_facilities.h" // por agora usamos estas definições da bib.
```

```
int main()
```

```
{
```

```
    cout << "Hello, world!\n";
```

```
    return 0;
```

```
}
```

*// notem o uso de ponto e vírgula; termina as instruções*

*// chavetas { ... } agrupam as instruções num bloco*

*// main() é uma função sem argumentos ()*

*// e tem como resultado um int (valor inteiro) para indicar sucesso ou //erro.*

# Hello, world!

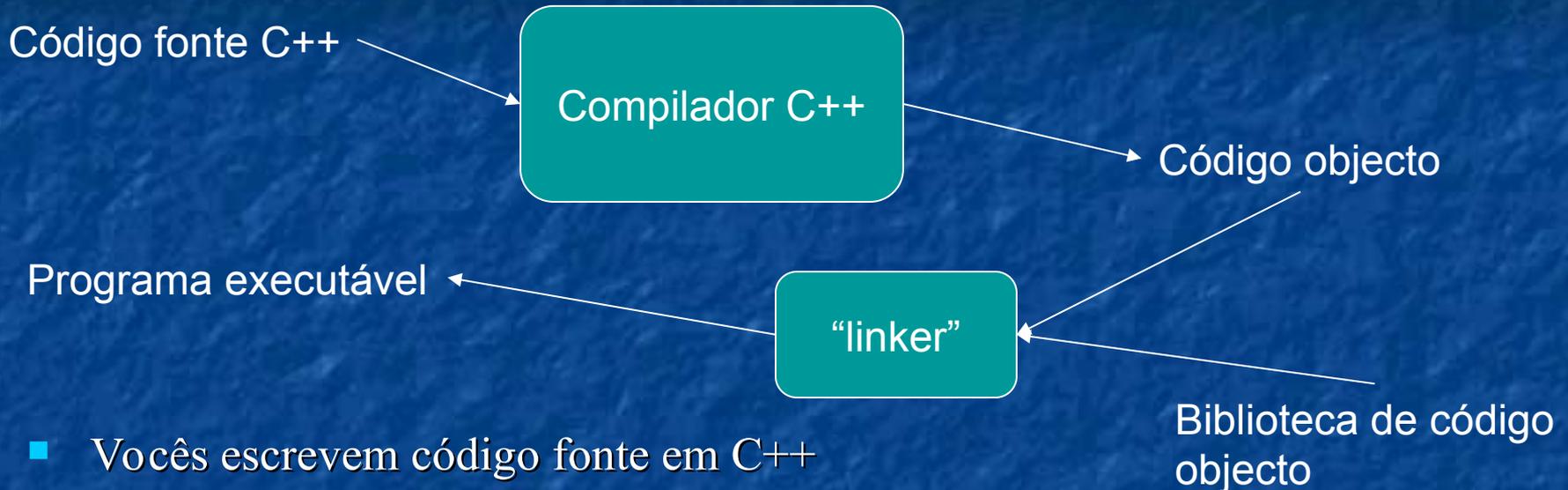
- “Hello world” é um programa muito importante
  - O seu objectivo é habituarem-se às ferramentas
    - Compilador
    - Ambiente de desenvolvimento
    - Ambiente de execução de programas
- Escrevam o programa cuidadosamente
  - Depois de estar a funcionar criem alguns erros para verem como respondem as ferramentas; por exemplo
    - Não incluam o cabeçalho
    - Não terminem a string
    - Escrevam mal **return** (e.g., **retrun**)
    - Esqueçam-se de um ponto e vírgula
    - Esqueçam-se de { }
    - ...

# Hello world

- É quase tudo código de suporte
  - Apenas `cout << "Hello, world!\n"` faz alguma coisa diretamente
- Isto é normal
  - A maior parte do nosso código, e a maioria dos sistemas que usamos, existem apenas para tornar outr código elegante e/ou eficiente
- “Código de suporte,” isto é, notação, bibliotecas, e outros suportes é o que torna o nosso código simples e eficiente.
  - Preferiam escrever 1,000,000 de linhas de código máquina?
- Isto implica que *não* devemos “fazer apenas as coisas”; devemos procurar garantir que o fazemos com elegância e correção e de forma a que facilite a criação de mais/outro software:

## Estilo é importante!

# Compilação e ligação



- Vocês escrevem código fonte em C++
  - O código fonte é (em princípio) legível
- O compilador traduz o que escrevem em código objecto (por vezes chamado código máquina)
  - O código objecto é compreensível para o computador (ao contrário do código fonte)
- O “linker” liga o vosso código ao código do sistema necessário para o executar
  - E.g., bibliotecas de I/O, código do SO e sistema de janelas
- O resultado é um programa executável.

# Então o que é a programação?

- Definições convencionais
  - Dizer a um idiota muito rápido exatamente o que fazer
  - Um plano para resolver um problema num computador
  - Especificar a ordem de execução de um programa
    - Mas os programas modernos envolvem milhões de linhas de código
    - E a manipulação de dados é fundamental
- Definições de outro domínio (academia)
  - Um ... programa é uma acumulação organizada e direcionada de recursos para atingir ... objetivos específicos
    - Boa, mas nenhuma menção a realmente fazer alguma coisa.
- A definição que usaremos
  - Especificação da estrutura e comportamento de um programa, e testar que o programa desempenha a sua tarefa de forma correcta e com desempenho aceitável
    - Nunca se esqueçam de verificar que “ele” trabalha
- Software == um ou mais programas

# Programar

- Programar é fundamentalmente simples
  - É só dizer o que máquina tem que fazer
- Então porque é que é difícil?
  - Nós queremos que “a máquina” faça coisas complexas
    - E os computadores são estúpidos, picuinhas e não perdoam erros
  - O mundo é mais complexo do que gostamos de admitir
    - Portanto nem sempre sabemos as implicações daquilo que pretendemos
  - “Programar é compreender”
    - Quando conseguem programar uma tarefa, compreendem-na
    - Quando programam, passam muito tempo tentando compreender a tarefa que querem automatizar
  - Programar é em parte prático, em parte teórico
    - Se forem apenas práticos, produzem “coisas” não escaláveis e sem manutenção possível
    - Se forem apenas teórico produzem apenas “brinquedos”

# A próxima aula

- Falaremos sobre tipo de dados, valores, variáveis, declarações, I/O simples, cálculos muito simples e segurança de tipos.