

# C++ : TO INFINITE AND BEYOND



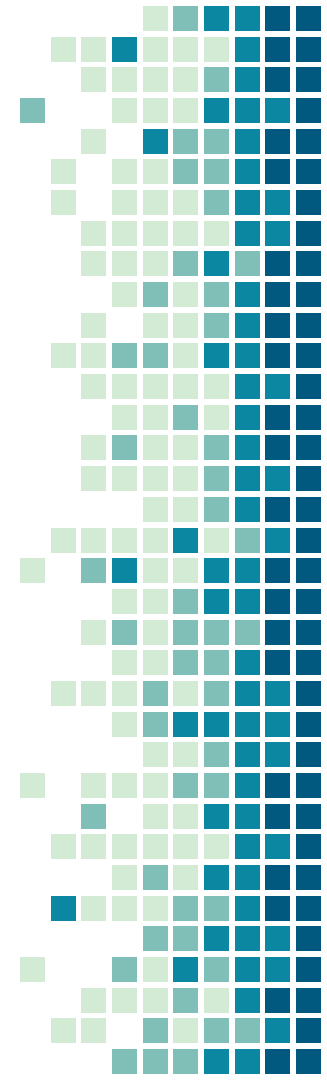


# \$[whoami]

**Caio Jordão Carvalho**

- StreamChord.
- KDE.
- LABRASOFT.

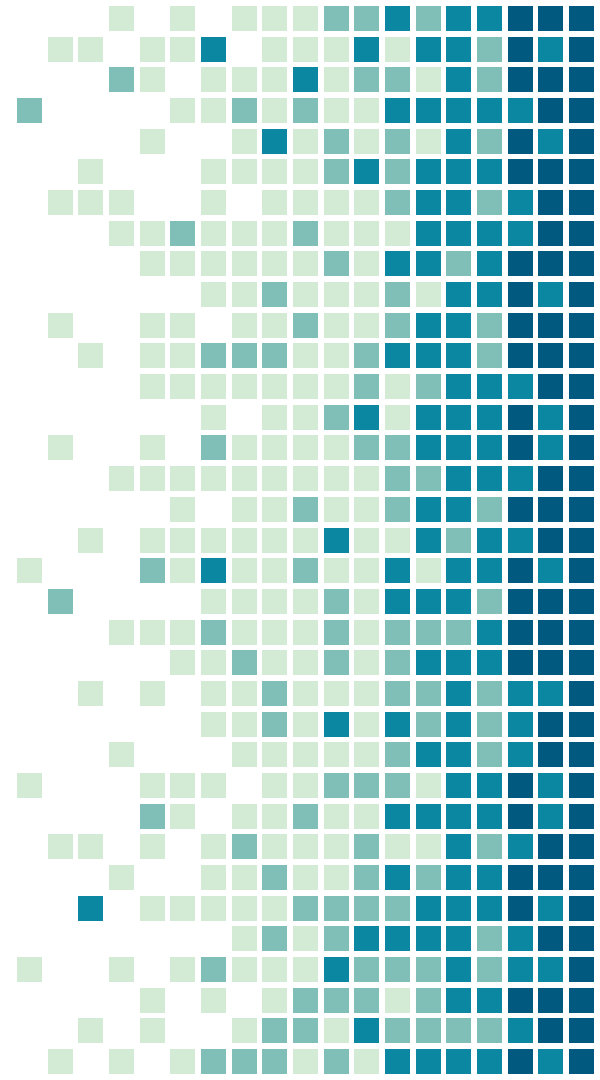
@cjlcarvalho



1.

# O que é C++?

Aprendendo um pouco sobre a origem...

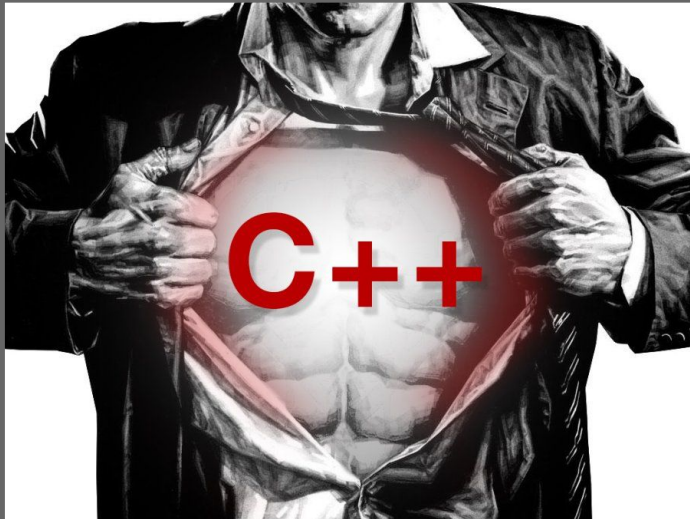


# C with Classes (1979)



- Bjarne Stroustrup.
- Multi-paradigma, procedural, funcional, orientada a objetos e genérica.
- Uso geral.

“ *Então é só um C  
melhorado?* ”



# Como eu conheci C++?

- Maratona de Programação. <3
- GRUPRO.
- Se já conheço C para valer, por que não tentar?



# 2.

## Recursos herdados do C

Em que C pode me ajudar no C++?

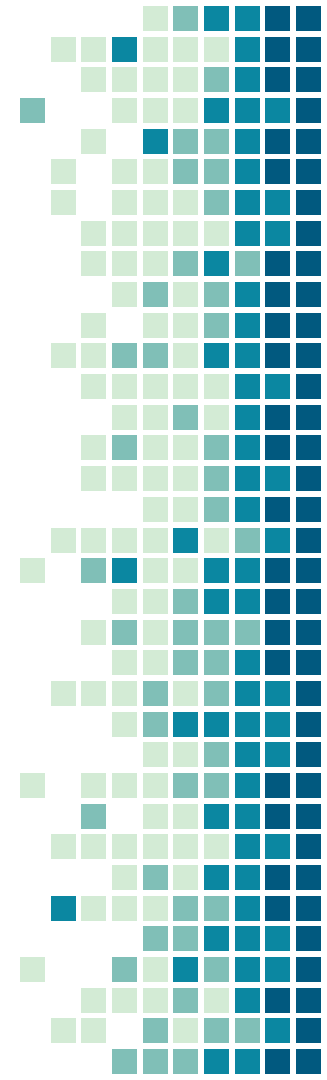


# Alguns desses recursos...

Compatibilidade com  
código em C.

In/Out padrão.

Alocação de memória  
(Ponteiros  
everywhere!!!).





# 3. Hello world!

Por que tudo tem um começo...



*pass by reference*



`fillCup( )`

*pass by value*



`fillCup( )`

[www.penjee.com](http://www.penjee.com)

4.

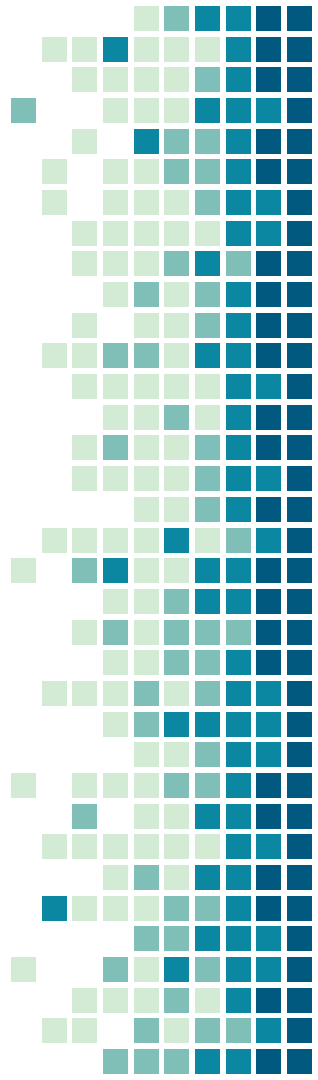
# Funções

Criando rotinas...



# Funções

- Passagem por valor, por ponteiro ou por operador de referência.
- Parâmetros default!
- Apontando para código: ponteiros de função.
- Sobrecarga.



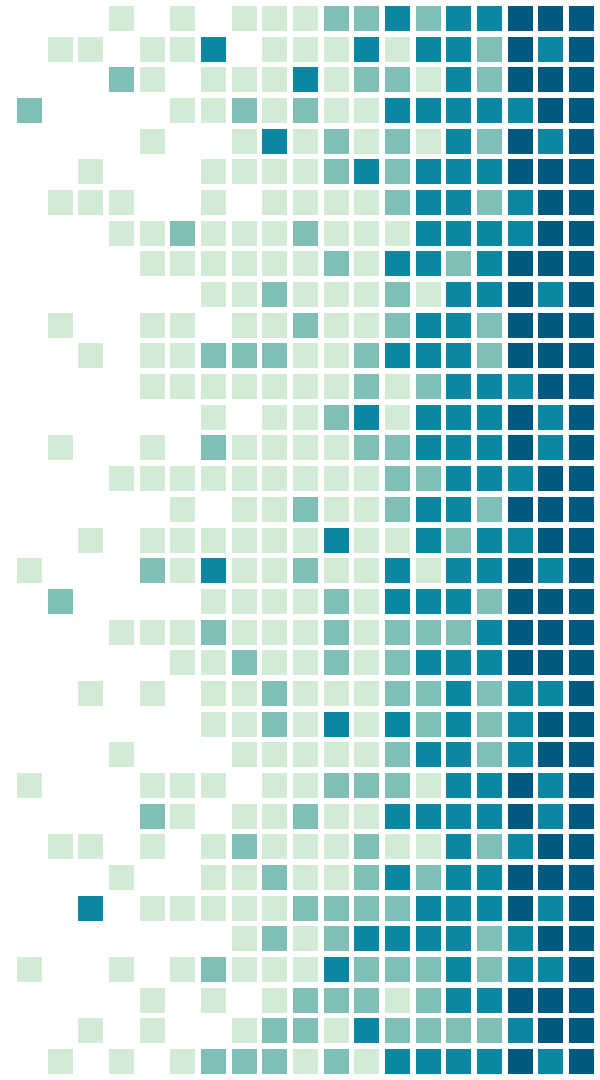
# Valor VS Ponteiro VS Op. de Referência

	Valor	Ponteiro	Op. Referência
Alocação	<b>Estática</b>	<b>Dinâmica</b>	<b>Estática</b>
Endereço mutável	<b>Não</b>	<b>Sim</b>	<b>Não</b>
Lifetime	<b>Escopo</b>	<b>Até ser liberado da memória</b>	<b>Escopo</b>
Valor mutável em função	<b>Não</b>	<b>Sim</b>	<b>Sim</b>

5.

# Orientação a Objetos

Criando Classes...



# Orientação a Objetos

- Struct.
- Class.
- Struct vs Class, qual usar?
- Headers: O que são? Onde vivem?
- Namespaces (o operador ::).



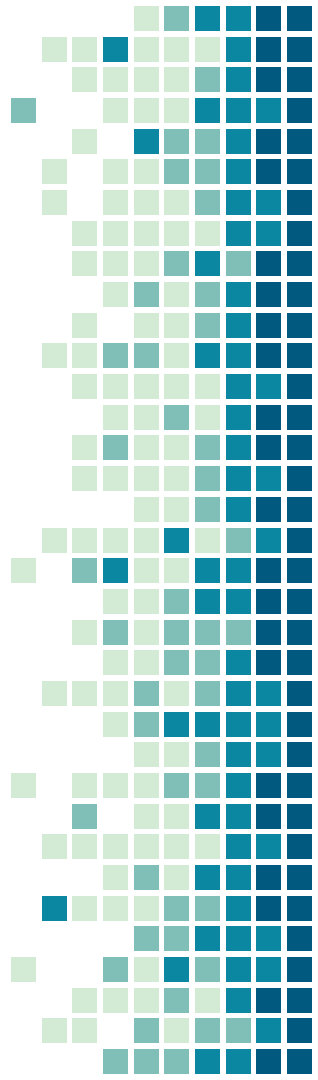
# Instanciando

- Alocação estática (você já conhece!).
- Alocação dinâmica (então o malloc era isso???).
- Objeto vs Referência de Objeto.



# Orientação a Objetos

- Construtores.
- Destrutores (GC é para os fracos).
- Memory Leak.
- O ponteiro this.
- Encapsulamento e modificadores de acesso.
- Funções const.





# Orientação a Objetos

- Membros estáticos (métodos e atributos).
- Classes e funções amigas (friend).
- Sobrecarregando operadores.
- Programação genérica.



# Orientação a Objetos

- Herança.
- Modos de herança.
- Herança múltipla (fuja do Diamond Problem).
- Métodos virtuais.
- Pure virtual methods.
- Deleted methods.

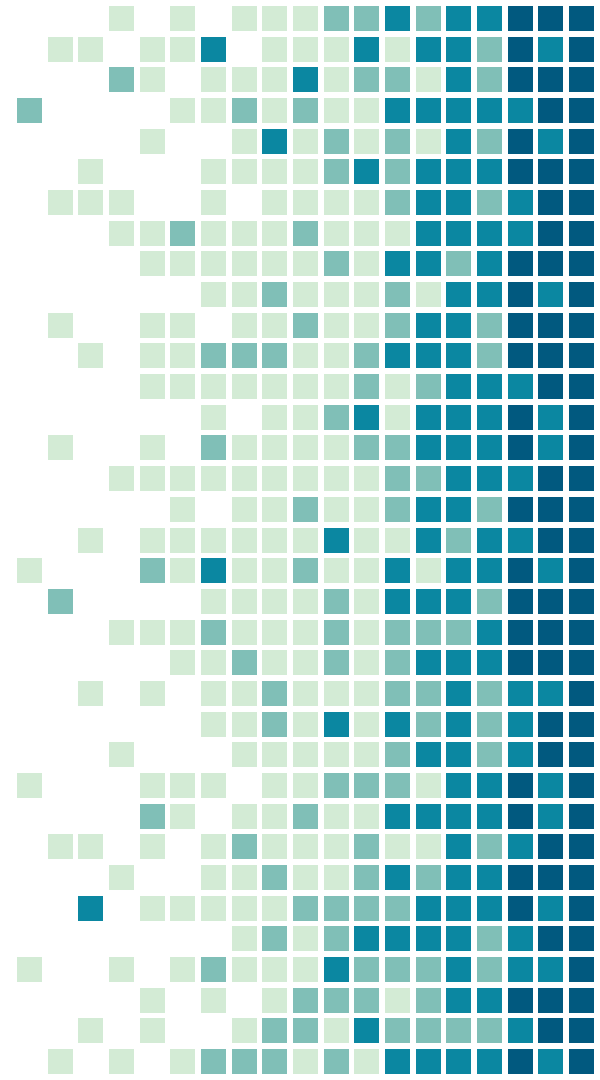




6.

O que é essa tal de  
STL?

Standard Template Library



# Standard Template Library

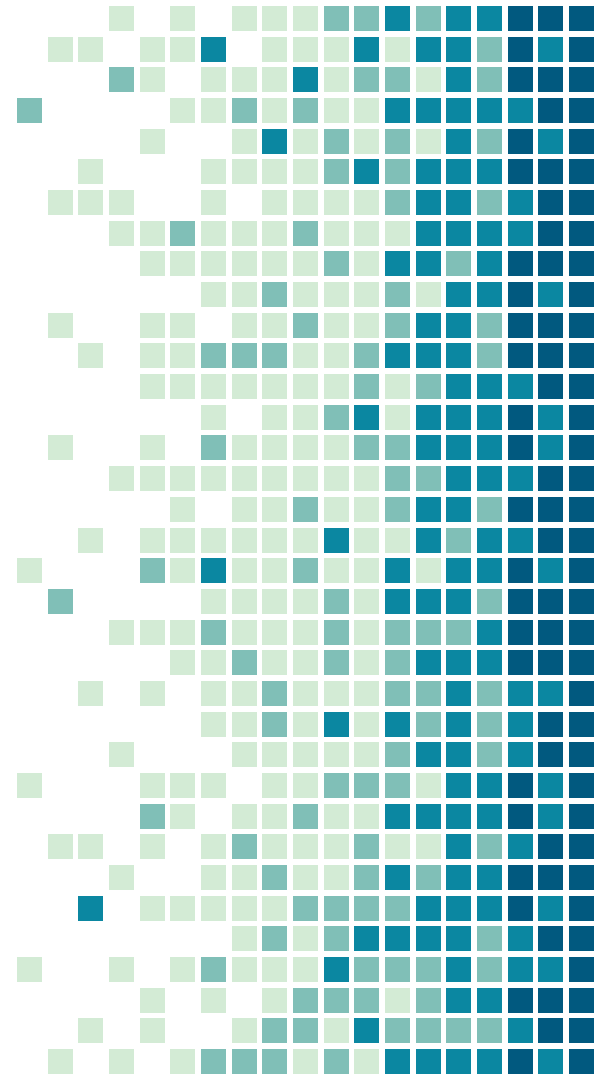
- STL != Biblioteca padrão do C++.
- “Biblioteca Padrão de Modelos”.
- Containers.
- Iterators.
- Algoritmos.
- Functors.



7.

Qt

Uma breve introdução a esse poderoso toolkit!



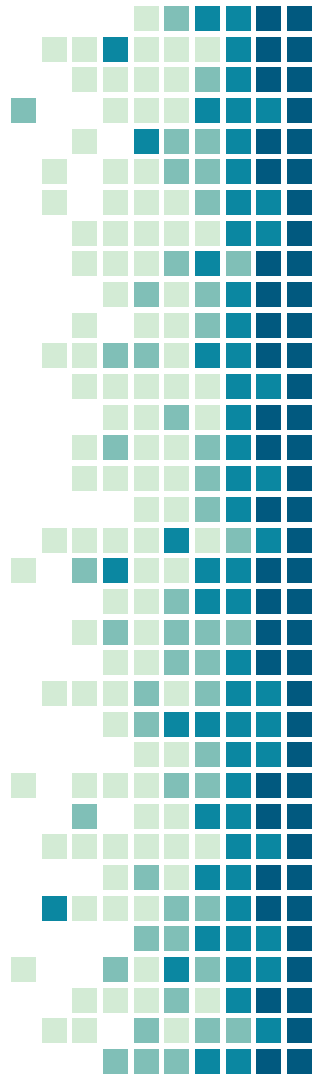
# Qt

- “cute”.
- Toolkit multiplataforma.
- Criado pela empresa Trolltech.
- Nokia.
- KDE.



# Qt

- Let's code...



8.

A melhor forma de  
estudar é...

All you need is code!





# Formas de estudar

- Software Livre.
- KDE.
- Maratona de Programação.
- Pelo menos uma linha de código por dia!



9.

E agora? Para onde  
devo ir?

A continuação da sua trilha pelo mundo  
do C++



# E agora, o que devo aprender?

- O compilador é seu melhor amigo. Leia as mensagens que ele te manda.
- C++11, C++14, C++17...
- Boost C++.
- Effective C++ (livro).



# FIM!

## Perguntas?

@cjlcarvalho

caiojcarvalho@gmail.com

caiojcarvalho.wordpress.com