

# Linguagem de Programação II

Prof. Antonio Carlos Sobieranski

DEC7532 | ENC | DEC | CTS



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA

# Apresentação

**Professor Antonio Sobieranski**

[a.sobieranski@ufsc.br](mailto:a.sobieranski@ufsc.br)

<https://lsim.ufsc.br>

**A316-JD / 206-MA**



Desde 2018 na UFSC

Disciplinas Atuais 2022.1: LP2-A, LP2-B, Tópicos Especiais

Coordenador do LSiIM – Laboratório de Sinais e Imagens Médicas – UFSC

Integrante INCoD – Instituto Nacional para Convergência Digital – UFSC

Colaborador LabTec – UFSC

## **Area de Concentração:**

- Processamento Digital de Imagens
- Reconhecimento de Padrões
- Visão Computacional
- Aprendizado de Máquina

# Plano de Ensino

Versão completa  
ver Moodle

	<b>Universidade Federal de Santa Catarina</b> Campus Araranguá - ARA Centro de Ciências, Tecnologias e Saúde Departamento de Computação PLANO DE ENSINO		
<b>SEMESTRE 2022.1</b>			
<b>I. IDENTIFICAÇÃO DA DISCIPLINA</b>			
<b>CÓDIGO</b>	<b>NOME DA DISCIPLINA</b>	<b>Nº DE HORAS-AULA SEMANAIS - TEÓRICAS</b>	<b>Nº DE HORAS-AULA SEMANAIS - PRÁTICAS</b>
DEC7532	Linguagem de Programação II	0	4
<b>TOTAL DE HORAS - AULA SEMESTRAIS</b>	<b>HORÁRIO TURMAS TEÓRICAS</b>	<b>HORÁRIO TURMAS PRÁTICAS</b>	<b>MODALIDADE</b>
72	02655A - 2.1620-2	02655A - 4.1620-2	Presencial
<b>II. PROFESSOR(ES) MINISTRANTE(S)</b>			
Prof. Antonio Carlos Sobieranski E-mail: <a href="mailto:a.sobieranski@ufsc.br">a.sobieranski@ufsc.br</a> Horário de atendimento: Terça-feira das 15:00 às 16:00 Local: por videoconferência ou local físico a ser definido e agendado com o professor.			
<b>III. PRE-REQUISITO(S)</b>			
DEC0012 Linguagem de Programação I			
<b>IV. CURSO(S) PARA O(S) QUAL(IS) A DISCIPLINA É OFERECIDA</b>			
ENGENHARIA DE COMPUTAÇÃO [Campus Araranguá]			
<b>V. JUSTIFICATIVA</b>			
O aluno ao final desta disciplina deverá ser capaz de transpor algoritmos, em consonância com linguagens de Programação I, para uma linguagem de programação que possibilite a implementação do paradigma de orientação à objetos.			
<b>VI. EMENTA</b>			
Fundamentos do paradigma Orientado à Objetos. Classes e Métodos. Encapsulamento, Herança e Polimorfismo. Modelagem e solução de problemas utilizando os conceitos de orientação a objetos, decomposição por objetos e tipos abstratos de dados. Interface gráfica com usuário (GUI). Análise dos aspectos tecnológicos complementares à solução de problemas (programação orientada a eventos, persistência de dados e objetos, tratamento de exceções).			
<b>VII. OBJETIVOS</b>			
<b>Objetivo Geral</b>			
<ul style="list-style-type: none"><li>Prover ao aluno conhecimentos do paradigma orientado à objetos de modo ao aluno ser capaz de transpor, modelar, implementar algoritmos.</li></ul>			
<b>Objetivos Específicos</b>			
<ul style="list-style-type: none"><li>Compreender a motivação para a adoção do paradigma de orientação por objetos.</li><li>Conhecer os principais pilares da orientação à objetos, bem como classes, abstração, herança e polimorfismo.</li><li>Decompor problemas segundo o conceito de orientação à objetos.</li><li>Implementar o conceito segundo os aspectos da orientação à objetos.</li></ul>			
<b>VIII. CONTEÚDO PROGRAMÁTICO</b>			
<b>UNIDADE 1:</b>			
<ul style="list-style-type: none"><li>Apresentação da disciplina. Apresentação do plano de ensino.</li><li>Da Linguagem C para C++ e Histórico</li><li>Utilização da biblioteca padrão do C++ e diferenças da programação em Linguagem C</li><li>Compilação, flags de compilação e depuração de código. Ferramentas para detecção de <i>Memoryleak</i> e <i>stackoverflow</i></li><li>Alocação dinâmica de memória em C++</li><li>Conceito de namespace, <code>std::string</code>, <code>stringstream</code>, <code>vector</code>, <code>pair</code>, <code>ifstream</code>, <code>ofstream</code>, and outros necessários da std.</li></ul>			
<b>UNIDADE 2:</b>			
<ul style="list-style-type: none"><li>Motivação para a adoção do paradigma de orientação por objetos.</li></ul>			

# Plano de Ensino

Portanto, as principais bases desta disciplina:



# Plano de Ensino

\*<https://github.com/halterman/CppBook-SourceCode>

## E-book\*

### Fundamentals of C++ Programming

**DRAFT**

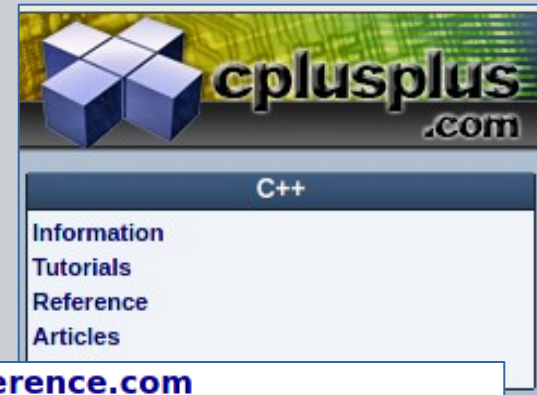
Richard L. Halterman  
School of Computing  
Southern Adventist University

February 5, 2019

## Livro BU



## Guias Referência



[cppreference.com](http://cppreference.com)

Page Discussion

## C++ reference

C++98, C++03, C++11, C++14, C++17, C++20, C++23

# Linguagem C++

## Histórico:

- Linguagem de programação de propósito geral (multi)
  - **Bjarne Stroustrup** na AT&T Bell Labs em meados de 1980.
- **Extensão** da Linguagem C da AT&T Bell Labs de 1970, ou "*C with Classes*".
- **Relação** com a Linguagem C:
  - C foi desenvolvido para escrever o Unix OS;
  - C amplamente usado para *system-level* software e sistemas embarcados;
  - C++ implementa características da POO;
  - C++ utiliza uma ampla coleção de código em C (recíproca inverdade).

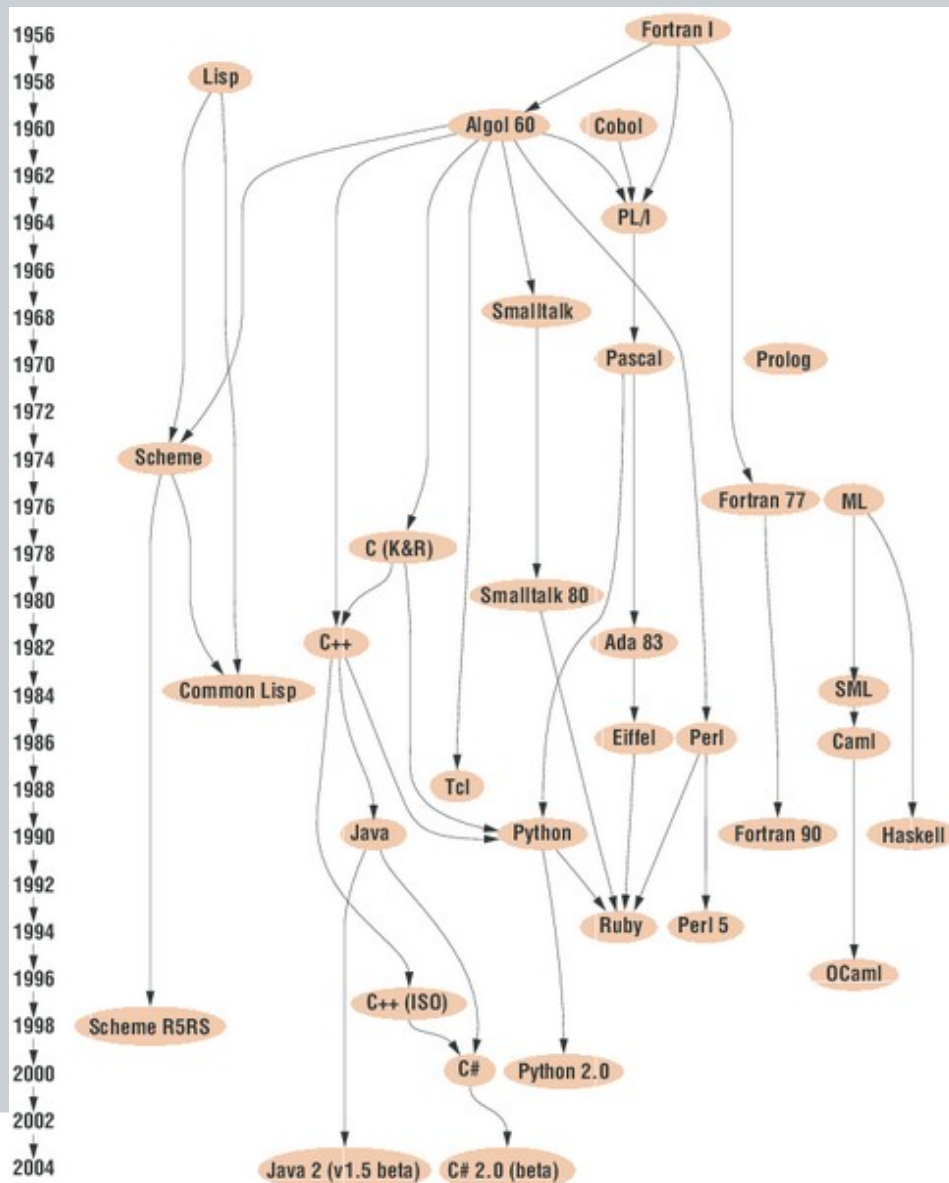


# Linguagem C++

## Histórico:

C++ standards

Year	C++ Standard <sup>[30]</sup>	Informal name
1998	ISO/IEC 14882:1998 <sup>[31]</sup>	C++98
2003	ISO/IEC 14882:2003 <sup>[32]</sup>	C++03
2011	ISO/IEC 14882:2011 <sup>[33]</sup>	C++11, C++0x
2014	ISO/IEC 14882:2014 <sup>[34]</sup>	C++14, C++1y
2017	ISO/IEC 14882:2017 <sup>[35]</sup>	C++17, C++1z
2020	ISO/IEC 14882:2020 <sup>[13]</sup>	C++20, C++2a

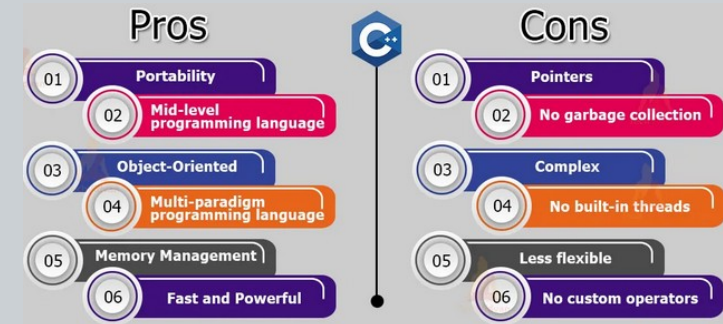




# Linguagem C++

## Vantagens e Desvantagens:

- (+) Portabilidade: Windows, Linux, MacOS
- (+) High / Mid-level
- (+) POO
- (+) Multi-paradigma: *Imperative, OO, Generic*
- (+) Gerência Memória
- (+) Performance / Poder
- (+) Similar a outras linguagens (*C#, C, and Java*)
- (+) Ampla variedade de aplicações (*OS, games, science*)

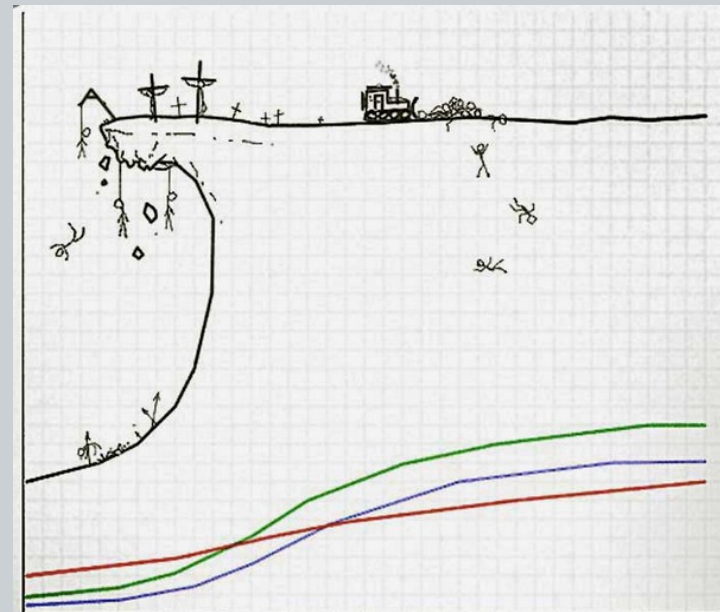
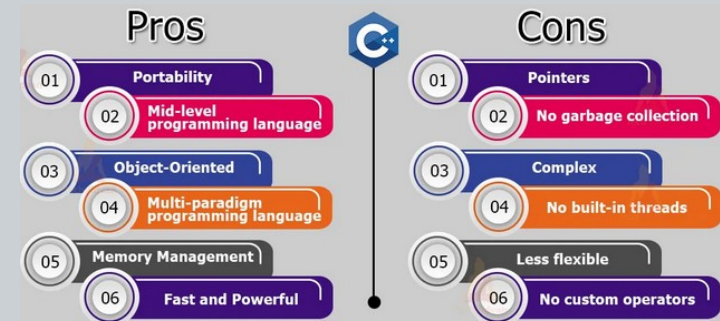




# Linguagem C++

## Vantagens e Desvantagens:

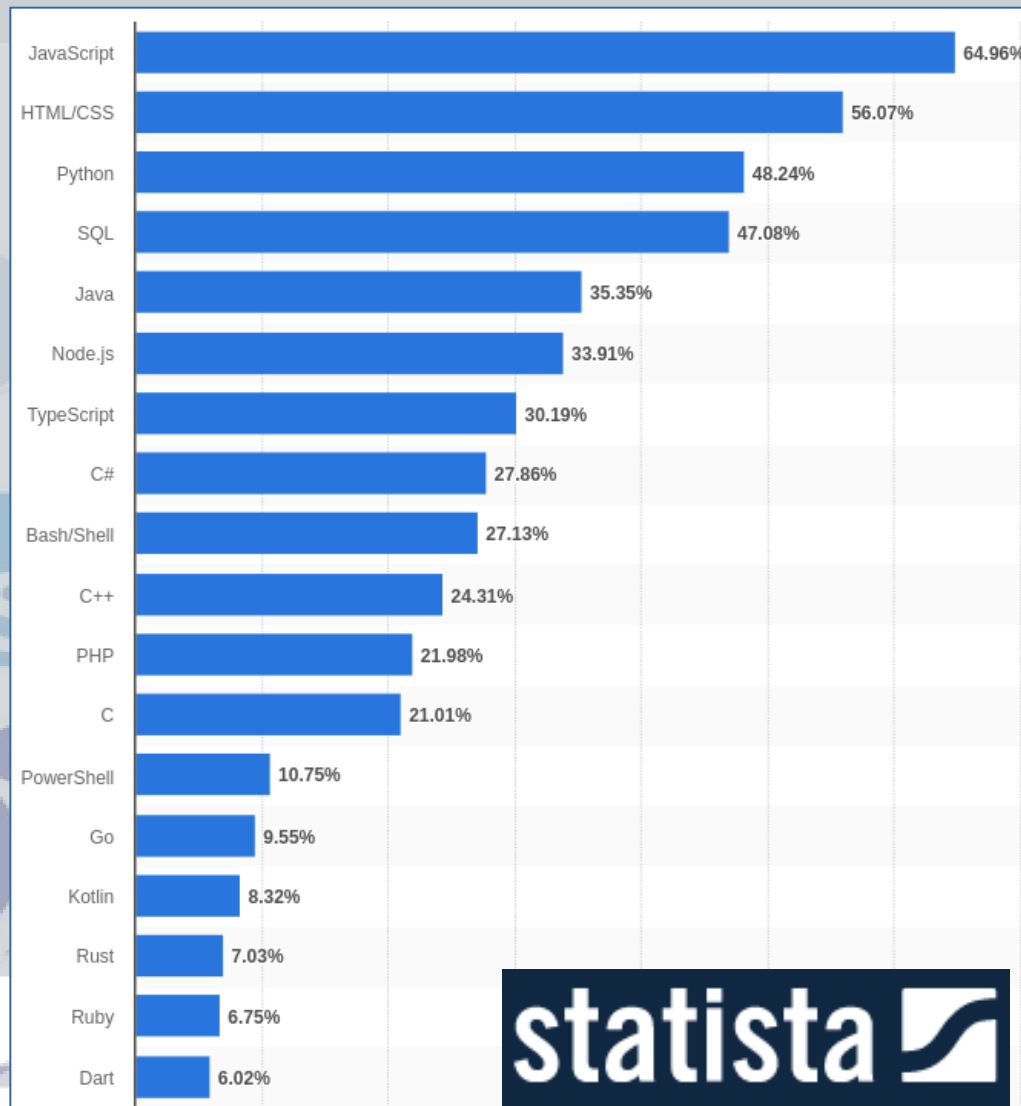
- (-) Ponteiros e seus cuidados
- (-) Insegura (?)
- (-) Complexa
- (-) *No built-in threads*
- (-) Carência de tipos de dados algébricos
- (-) Maior curva de aprendizado





# Linguagem C++

***Most used programming languages among developers worldwide, as of 2021***



# Linguagem C++

## Estrutura Geral

```
include directives  
  
int main() {  
    program statements  
}
```

### Listing 2.1: simple.cpp

```
#include <iostream>  
  
int main() {  
    std::cout << "This is a simple C++ program!\n";  
}
```

This is a simple C++ program!

# Linguagem C++

## Editando, Compilando e Executando o Programa

Programadores em C++ possuem 2 formas de criar os programas:

- Ambiente em linha de comando: *CLI – command line interface*)

Terminal em Linux / MacOS

CMD em windows

```
sobieranski@localhost ~ $ g++  
g++: fatal error: no input files  
compilation terminated.  
sobieranski@localhost ~ $
```

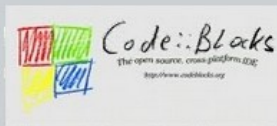
- Utilizando *IDE – Integrated Development Environment*



NetBeans



C/C++ IDE



Visual  
Studio



VSCode

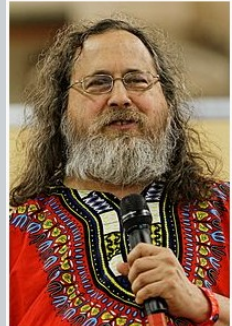
# Linguagem C++

## Editando, Compilando e Executando o Programa

- Compiladores



Richard Stallman



# Linguagem C++

## Sugestão de Estudos

- E-book base da disciplina – Ler até página 14 (introdução desta aula)
- E-book base da disciplina – Ler páginas 15 – 288 (Cap.10) (suave imersão na Linguagem C++ a partir da Linguagem C) (contém alguns poucos elementos novos do C++, mas sem prejuízo ao aluno)
- Esta disciplina continua nas próximas aulas a partir do Cap.11



## Contato

Prof. Antonio Carlos Sobieranski – DEC | A316JD / 206MA

E-mail: [a.sobieranski@ufsc.br](mailto:a.sobieranski@ufsc.br)

<https://lsim.ufsc.br>



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA