

# ACH-2001 – Introdução à Programação

Marcos Lordello Chaim

Bacharelado em Sistemas de Informação (BSI)  
Escola de Artes, Ciências e Humanidades (EACH)  
Universidade de São Paulo (USP)

# Disciplina

## Objetivo:

- ▶ Introduzir o aluno aos conceitos básicos de computação e lógica de programação, sob o paradigma imperativo.

# Disciplina

## Objetivo:

- ▶ Introduzir o aluno aos conceitos básicos de computação e lógica de programação, sob o paradigma imperativo.

## Não é um curso de sobre a linguagem C!

- ▶ Trata-se de um curso de programação que utiliza a linguagem C
- ▶ Por que C?
  - ▶ Linguagem imperativa bastante usada
  - ▶ Permite que se aprenda detalhes sobre a alocação e o acesso à memória.

# Método de Ensino

## Incremental:

- ▶ Começaremos com noções mais básicas, indo até a criação de programas bem estruturados
- ▶ Ao final do curso, você não irá reconhecer os programas do início
- ▶ Por isso os programas iniciais não serão exemplos de boa programação

# Método de Ensino

## Indução ao erro:

- ▶ Errar é uma das melhores formas de aprender
- ▶ Implica experimentar, e analisar os resultados
- ▶ Em vários casos apresentaremos programas propositalmente errados, para que vocês possam ver o comportamento do sistema e identificar a razão do erro

# Disciplina

## A quem então se destina o curso?

- ▶ A quem não conhece sequer o significado de “programar”
- ▶ Quem tem alguma noção aproveita também, e bastante

## Do que precisaremos para o curso?

- ▶ De um compilador para a linguagem C (MingW, Cygwin, gcc ou um compilador online)
- ▶ Há vários compiladores online, por exemplo:  
<https://replit.com/languages/c>

# Material

## Como instalar no Windows:

- ▶ Você deve baixar um dos compiladores disponíveis compatíveis com a sua versão do sistema operacional.
- ▶ <http://linguagemc.com.br/ides-e-compiladores-c/>

# Material

## Como instalar no Linux (Ubuntu):

- ▶ Tipicamente gcc já está instalado no Linux.
- ▶ Mas pode ser instalado utilizando o seguinte comando `sudo apt-get install gcc`:
- ▶ <https://terminaldeinformacao.com/2013/08/13/como-utilizar-o-gcc-no-linux/>



# Material

## Usaremos uma IDE?

- ▶ Integrated Development Environment – ambiente integrado para desenvolvimento de software (editor, compilador, depurador etc)
- ▶ Há várias opções que podem ser usadas...
- ▶ E qual usar?

# Material – IDEs

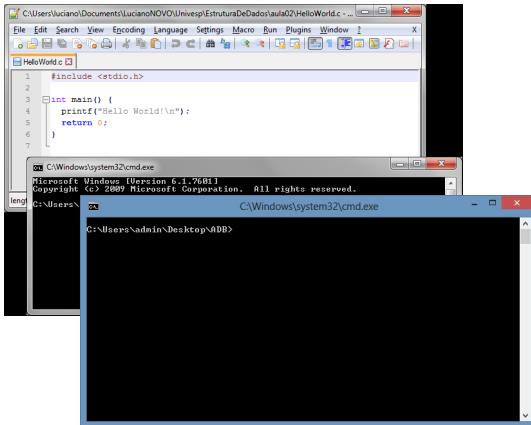
## IDEs para a linguagem C:

- ▶ Code Blocks: <https://www.codeblocks.org/>
- ▶ Dev-C++: <https://bloodshed.net/>

# Material – IDEs

E o que assumirei que vocês têm?

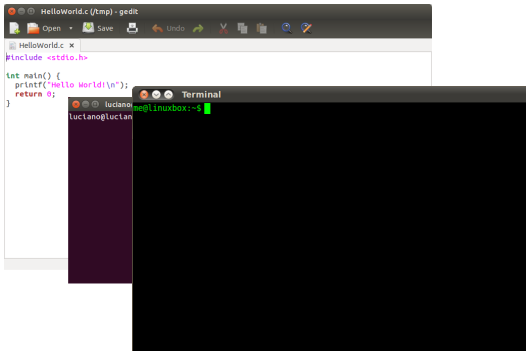
Notepad++



# Material – IDEs

E o que assumirei que vocês têm?

Ou Gedit



# Material – IDEs

## Por que não usaremos uma IDE?

- ▶ Melhor modo de se entender o que acontece em nosso programa
  - ▶ Não há interferência de nada, tentando nos “ajudar”
- ▶ Não há a carga cognitiva exigida para aprender a usar a IDE

# Material de Apoio

## Listas de Exercício

- ▶ Em conjunto com as aulas teremos algumas listas com exercícios
- ▶ É de extrema importância que sejam feitas, pois acompanham o conteúdo, aumentando o grau de dificuldade a cada exercício
- ▶ Não esqueçam que programação não é uma disciplina teórica
  - ▶ Somente a prática faz um bom programador

# Material de Apoio

Há também bons cursos via web:

Video aulas do professor Luciano Digiampietri:

[https://www.youtube.com/playlist?list=PL\\_JAaU8k6DQWsh1mt8vwpP8YsH6Xhgq2N](https://www.youtube.com/playlist?list=PL_JAaU8k6DQWsh1mt8vwpP8YsH6Xhgq2N)

# Programação de aulas – Março

<b>Data</b>	<b>Tópico</b>
20/03/2023	Apresentação da disciplina
24/03/2023	Problemas e algoritmos; computadores
27/03/2023	Sistema Operacional e Linguagens de Programação Primeiro programa
31/03/2023	Um Primeiro Programa (continuação)



## Programação de aulas – Abril

<b>Data</b>	<b>Tópico</b>
03/04/2023	Feriado
07/04/2023	Feriado
10/04/2023	Variáveis
14/04/2023	Funções
17/04/2022	Condicionais (parte 1)
21/04/2023	Feriado
24/04/2023	Condicionais (parte 2)
28/04/2023	Operadores Lógicos

## Programação de aulas – Maio

<b>Data</b>	<b>Tópico</b>
01/05/2023	Feriado
05/05/2023	Condicionais (parte 3)
08/05/2023	Laços (parte 1)
12/05/2023	Laços (parte 2)
15/05/2023	Laços (parte 3)
19/05/2023	Exercícios
22/05/2023	Prova 1 (P1)
26/05/2023	Alocação de Memória
29/05/2023	Arranjos (parte 1 e parte 2)

## Programação de aulas – Junho

<b>Data</b>	<b>Tópico</b>
02/06/2023	Arranjos (parte 3) Entrega do Exercício-Programa 1 (EP1)
05/06/2023	Caracteres
08/06/2023	Feriado
12/06/2023	Matrizes (parte 1 e parte 2)
16/06/2023	Estruturas (parte 1)
19/06/2023	Estruturas (parte 2)
23/06/2023	Leitura da Entrada
26/06/2023	Busca Sequencial e Binária
30/06/2022	Ordenação

## Programação de aulas – Julho

<b>Data</b>	<b>Tópico</b>
03/07/2023	Lista ligada
07/07/2023	Exercícios Entrega do Exercício-Programa 2 (EP2)
10/07/2023	Prova 2 (P2)
14/07/2023	Prova substitutiva (Psub)
17/07/2023	Prova de recuperação (Prec)

# Informações de contato

- ▶ Professor: Marcos Lordello Chaim
- ▶ `chaim@usp.br`
- ▶ sala I1-322N
- ▶ Horário de atendimento: Segunda-feira das 13h30-14h30.  
Necessário agendar.

# Regras do Jogo

## Site da disciplina: Moodle

- ▶ <https://edisdisciplinas.usp.br/>
- ▶ Acessando sua conta no Moodle:
  - ▶ Acessem o ambiente usando o mesmo login e senha única dos sistemas USP
  - ▶ Lá vocês encontrarão todas as disciplinas das quais participam

# Regras do Jogo

- ▶ A programação de aulas é preliminar e pode ser alterada no decorrer do semestre.
- ▶ Aulas presenciais.
- ▶ Uso do e-disciplinas. As comunicações enviadas aos alunos via o e-disciplinas são assumidas como de conhecimento de todos.
- ▶ Frequência obrigatória de 70%.
- ▶ Avaliação será por meio de provas e EPs.
- ▶ EPs serão individuais. Ferramentas para detecção de plágio serão utilizadas.

# Critério de Avaliação

- ▶ Duas provas: P1 e P2 (Psub substitui ou P1 ou P2)
- ▶ Dois Exercícios Programados (EPs): EP1 e EP2.
- ▶  $M_{Provas} = \frac{P1 + 2 \cdot P2}{3}$
- ▶  $M_{EPs} = \frac{EP1 + EP2}{2}$  (média aritmética)
- ▶ Se  $M_{EPs} \geq 5,0$  e  $M_{Provas} \geq 5,0$ 
  - ▶  $\Rightarrow$  Média Final =  $0,4 \times M_{EPs} + 0,6 \times M_{Provas}$
- ▶ Caso contrário,
  - ▶  $\Rightarrow$  Média Final =  $\text{Mínimo}(M_{Provas}, M_{EPs})$



# Critério de avaliação

## Resultado:

- ▶ Se Frequência  $< 70\%$   $\rightarrow$  reprovado
- ▶ Caso contrário (Frequência  $\geq 70\%$ ):
  - ▶ Se Média Final  $\geq 5,0$   $\rightarrow$  aprovado
  - ▶ Se  $3,0 \leq$  Média Final  $< 5,0$   $\rightarrow$  recuperação
  - ▶ Se Média Final  $< 3,0$   $\rightarrow$  reprovado

# Critério de avaliação

## Recuperação:

- ▶ Prova (Prec) sobre toda matéria da disciplina.
- ▶ Apenas para quem obteve:
  - ▶ Frequência  $\geq 70\%$  e
  - ▶  $3,0 \leq \text{Média Final} < 5,0$ .
- ▶ Média final recuperação ( $M_{FR}$ ) é calculada:
  - ▶  $M_{FR} = 0,5 \times \text{Média Final} + 0,5 \times \text{Prec}$
- ▶ Se  $M_{FR} \geq 5,0 \rightarrow$  aprovado.
- ▶ Se  $M_{FR} < 5,0 \rightarrow$  reprovado.

# Bibliografia

1. PAES, R.B. Introdução à Programação com a Linguagem C. Novatec. 2017
2. SHILDT, H. C completo e total. 3a. ed. Makron. 1997.

# Finalizando

Dúvidas?