



# Apresentação da Disciplina de Programação de Computadores I



Leonardo Murta

leomurta@ic.uff.br

# Apresentações

---

- ▶ Quem sou eu?

- ▶ Leonardo Murta

- ▶ <http://www.ic.uff.br/~leomurta>

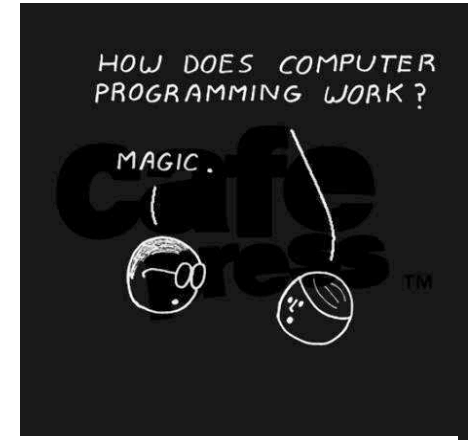
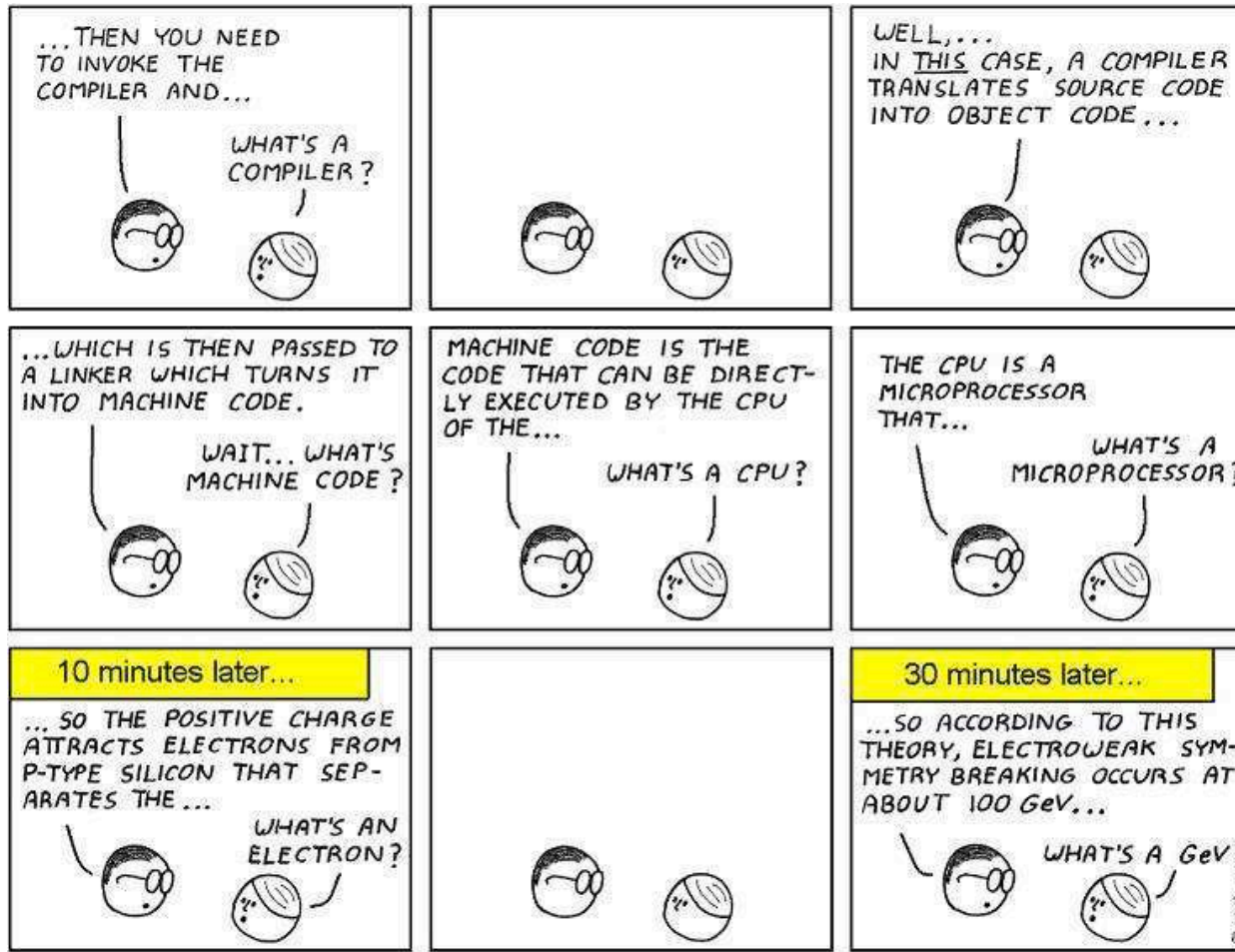
- ▶ Quem são vocês?

- ▶ Nome?

- ▶ Onde estudou? O que sabe de computação?

- ▶ Algum hobby “sério”?

# O quê vocês esperam do curso?



# Aulas

---

- ▶ **Segundas e Quartas**
  - ▶ Sala 321
- ▶ **Laboratório**
  - ▶ Solicitamos ao IC/UFF e estamos aguardando a resposta
- ▶ **Estágio em Docência**
  - ▶ Rubén Interian Kovaliova

# Site do Curso

Leiam as **regras** do curso no site, anotem as **datas** e tragam as dúvidas na próxima aula



**Leonardo Gresta Paulino Murta**  
Assistant Professor (Professor Adjunto IV), IC/UFF  
D.Sc., COPPE/UFRJ, 2006  
M.Sc., COPPE/UFRJ, 2002  
B.Sc., IM/UFRJ, 1999



## Programação de Computadores I

### Logística

Disciplina: TCC00308 - Programação de Computadores I (turma F-1)

Data: segundas e quartas, de 18:00 às 20:00

Sala: 321

Sala virtual: usaremos o **Google Classroom** como nossa sala de aula virtual. Lá serão divulgadas listas de exercícios valendo nota e por lá devem ser feitas as entregas. O **Google Classroom** será usado também para divulgar avisos gerais e para dúvidas.

Importante: todos os alunos devem se inscrever no **Google Classroom** na primeira semana de aula. Para se inscrever, é necessário ter uma conta no UFF Mail ([ic@uff.br](mailto:ic@uff.br)), clicar no símbolo de "+" no canto superior direito da página do **Google Classroom**, e selecione a opção "Participar da Turma". O código de inscrição na turma será fornecido no primeiro dia de aula.

### Avaliação

$Média = (2 \times Prova 1 + 2 \times Prova 2 + Lista de Exercícios) / 5$

#### APROVADO

(Presença  $\geq 75\%$ ) E (Média  $\geq 6$ )

#### VERIFICAÇÃO SUPLEMENTAR

(Presença  $\geq 75\%$ ) E ( $4 \leq Média < 6$ )

Será aprovado na VS se tirar nota maior ou igual a 6.

#### REPROVADO

(Presença  $< 75\%$ ) OU (Média  $< 4$ )

### Estagiário de Docência

Rubén Interian Kovaliova (enviar dúvidas no **Google Classroom**)

### Lista de exercícios

Ao longo da disciplina, serão realizados diversos exercícios. Esses exercícios comporão a média. Todos terão data de entrega definida e deverão ser entregues pelo **Google Classroom** da disciplina.

### Ferramentas

O aluno tem liberdade de utilizar a IDE Python de sua preferência. Contudo, nas atividades de sala de aula será utilizado a **IDE PyCharm**.

- Interpretador Python
- IDE PyCharm

### Bibliografia

Paul Barry, Use a Cabeça! Programação, Editora Alta Books.

Paul Barry, Use a Cabeça! Python, Editora Alta Books.

Furlan, M., Gomes, M., Soares, M., Concílio, R., 2005, Algoritmos e Lógica de Programação, Editora Thomson.

### Cronograma

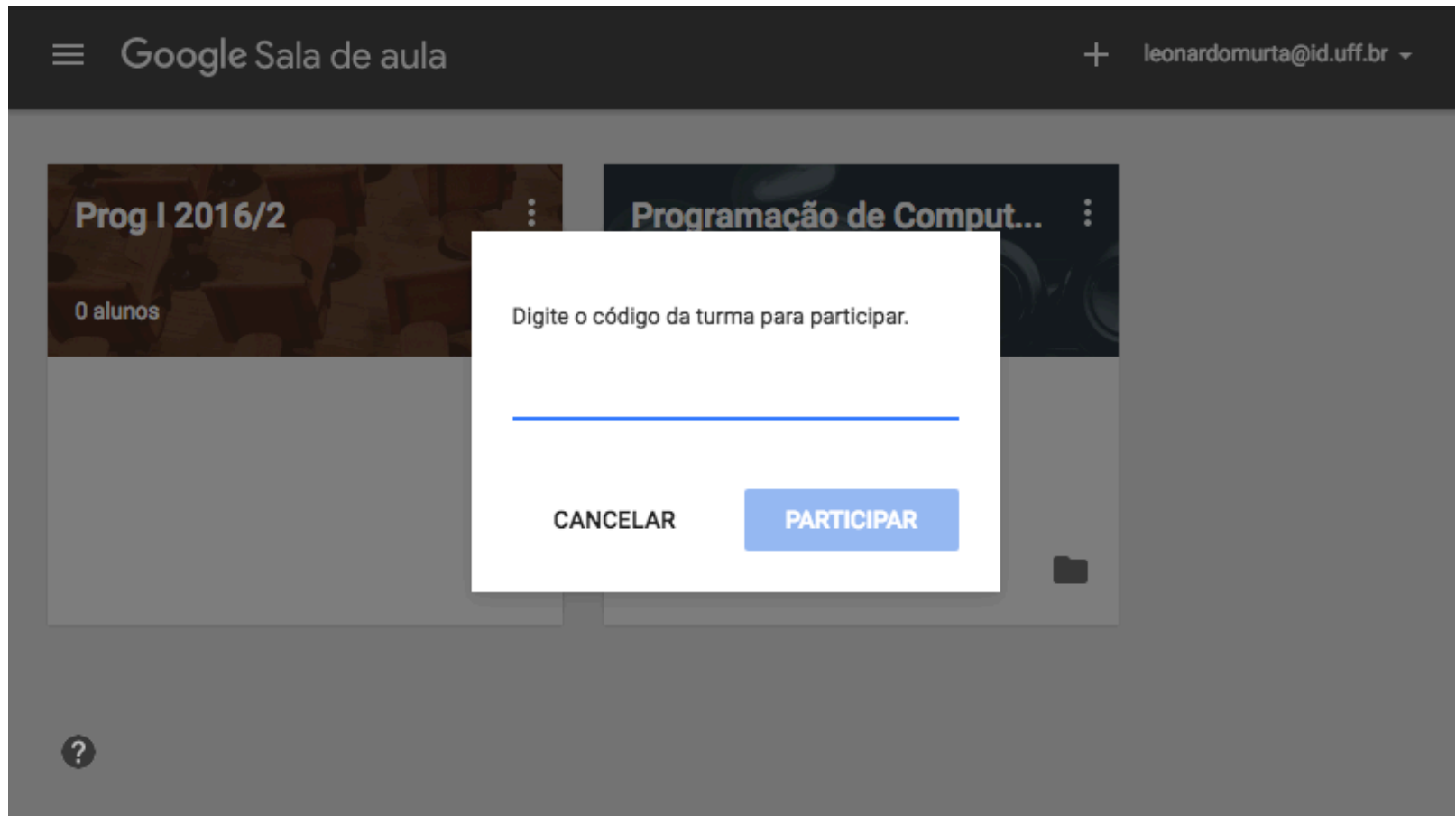
Data	Atividade
29/08/2016	Aula
31/08/2016	Aula

**Importante:** cadastrem-se no Google Classroom!



# Sala Virtual

---



<http://classroom.google.com>

# Objetivos da Disciplina

---



- ▶ Solucionar problemas (x 1000)
- ▶ Desenvolver pensamento computacional
- ▶ Escrever e ler na linguagem do computador
- ▶ **Objetivo secundário: Programar em Python**
  - ▶ Atualmente a mais popular linguagem introdutória de cursos de programação nas universidades top dos EUA
  - ▶ Criada por Guido van Rossum, por volta de 1991
  - ▶ Fácil partir para outras linguagens, se necessário

# Quem usa Python?

---





# Avaliação

---

$$Média = \frac{2 \times Prova_1 + 2 \times Prova_2 + Exercícios}{5}$$

# Avaliação

---

- ▶ **APROVADO:** (Presença  $\geq 75\%$ ) E (Média  $\geq 6$ )
- ▶ **VS:** (Presença  $\geq 75\%$ ) E ( $4 \leq$  Média  $< 6$ )
  - ▶ Será aprovado na VS se tirar nota maior ou igual a 6
- ▶ **REPROVADO:** (Presença  $< 75\%$ ) OU (Média  $< 4$ )

# Exercícios

---

- ▶ Serão apresentados exercícios em todas as aulas
- ▶ Alguns serão feitos em sala, os demais devem ser feitos em casa
- ▶ Todos terão data para serem entregues e devem ser entregues via Google Classroom
- ▶ Os exercícios, feitos individualmente, serão corrigidos pelo estagiário de docência e a nota comporá a média

# *Fair Play!*

---

- ▶ Não colar ou dar cola em provas
- ▶ Não plagiar o trabalho
- ▶ Não trapacear nos trabalhos e listas de exercício
- ▶ Não sobrecarregar os colegas do grupo
- ▶ Não assinar presença por colegas
- ▶ Dar crédito apropriado quando usar trabalhos de terceiros



<http://www.claybennett.com/pages/ethics.html>



# Dois conselhos

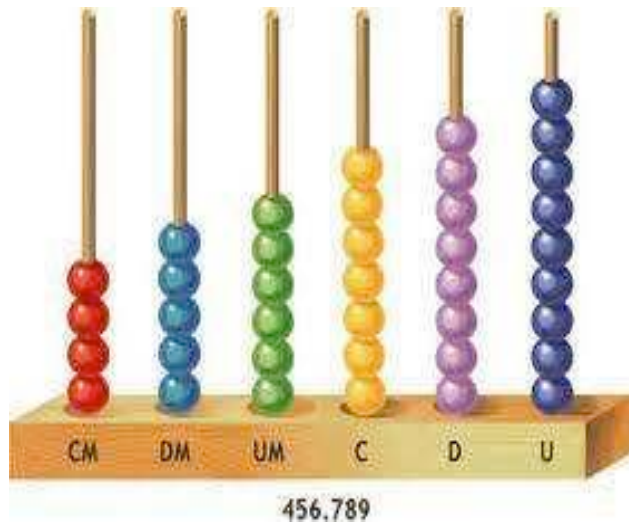
---

1. Aprender a programar é como aprender a tocar um instrumento musical: não basta ler, **tem que praticar**
2. Escreva seus programas de forma que seja fácil para outras pessoas entenderem

# Um pouco de história

## ▶ 2400 AC: Ábaco

- ▶ Primeira tentativa de se criar um artefato capaz de contar



## ▶ 1642: Pascalina

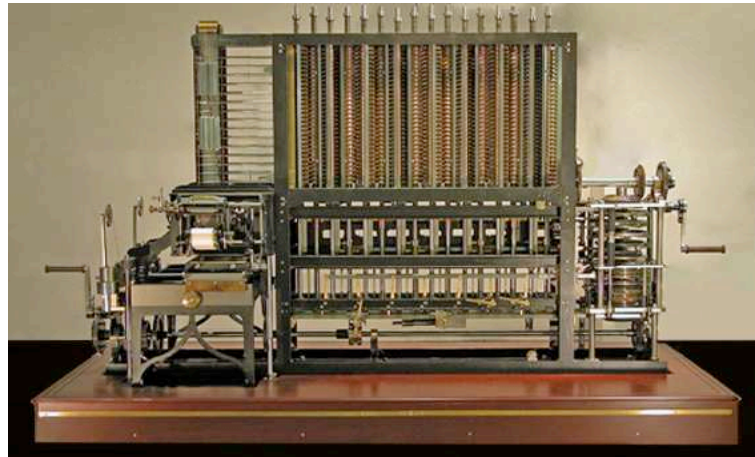
- ▶ Criada por Blaise Pascal, aos 19 anos
- ▶ Uma das primeiras máquinas mecânicas de calcular



# Um pouco de história

---

- ▶ **1822: Primeiro computador mecânico**
  - ▶ Projetado por Charles Babbage mas não terminado devido à falta de recursos
  - ▶ Posteriormente, o seu projeto foi construído e exposto em um museu



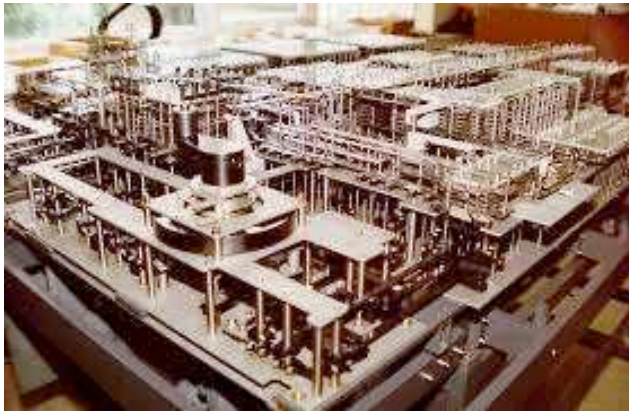


# Um pouco de história

---

## ▶ 1938: ZI

- ▶ O primeiro computador programável
- ▶ Muito foi perdido por causa da II Guerra Mundial



## ▶ 1943: Colossus

- ▶ Usado pelos britânicos para decodificar mensagens alemãs



# Um pouco de história

---

## ▶ 1946: ENIAC

- ▶ Considerado o primeiro computador eletrônico de propósito geral
- ▶ Construído na Universidade da Pensilvânia
- ▶ Ocupava uma sala inteira
- ▶ Pesava 30 toneladas
- ▶ Consumia 200 kw de potência
- ▶ Entrada: leitora de cartões perfurados
- ▶ Saída: perfuradora de cartões



ENIAC  
<http://www.upenn.edu>

# Um pouco de história

---

- ▶ **1973:Alto**
  - ▶ Primeiro computador pessoal
  - ▶ Construído pela Xerox, mas nunca produzido em massa
  - ▶ Já tinha mouse, interface gráfica e sistema operacional
  - ▶ As características do Alto foram incorporadas no Macintosh



# Um pouco de história

---

- ▶ Hoje: Computação móvel
- ▶ Laptop, Tablet, Celular
- ▶ Entrada: teclado, tela
- ▶ Saída: tela



# Hardware x Software

---

## Hardware

- ▶ Peças
- ▶ Corpo



## Software

- ▶ Programas
- ▶ Alma



# Computadores e programas

---

- ▶ Computador: máquina que pode executar programas
- ▶ Programa: sequência bem precisa de passos que um computador deve executar
- ▶ Linguagem de programação: linguagem projetada para produzir programas de computadores





# O que os computadores entendem?

---

- ▶ Para que o computador faça o que você quer (e ele pode fazer quase tudo que você mandar), **é necessário falar a linguagem dele**
- ▶ Qual é a linguagem que o computador fala?

# Bits and pieces

---

“Your computer successfully creates the illusion that it contains photographs, letters, songs, and movies. All it really contains is bits, lots of them, patterned in ways you can't see. **Your computer was designed to store just bits - all the files and folders and different kinds of data are illusions created by computer programmers.**”

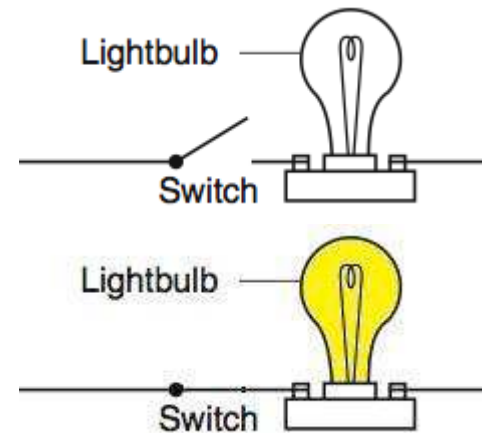
(Hal Abelson, Ken Ledeen, Harry Lewis, in "Blown to Bits")



# Instruções no computador

---

- ▶ Operações no computador são grupos de bits
  - ▶ 0 ou 1
  - ▶ Desligado ou ligado
- ▶ **Microprocessador**
  - ▶ Move o conteúdo de grupos de bits
  - ▶ Soma pares de grupos de bits
  - ▶ Subtrai um grupo de bits de outro
  - ▶ Compara pares de grupos de bits
  - ▶ ...



# Bits and pieces

---

“There are only 10 different kinds of people in the world:  
those who know binary and those who don't.”  
- Anônimo

- ▶  $1234 = 10011010010$
- ▶ “A” = 65 (decimal) = 01000001

# Linguagens de montagem

---

- ▶ Usa símbolos mais amigáveis aos seres humanos para representar as instruções
- ▶ A memória do computador e os registradores também recebem nomes simbólicos
- ▶ Não são bits, mas ainda requerem um nível de detalhe muito próximo ao que a máquina de fato faz

# Assembler – Soma de dois números (17 e 5)

---

```
main PROC
    mov eax, offset x
    push eax
    mov eax, 17
    push eax
    mov eax, 5
    push eax
    pop ebx
    pop eax
    add eax, ebx
    push eax
    pop eax
    pop ebx
    mov [ebx], eax
    call writeint
    call crlf
    exit
main ENDP
END main
```

# Linguagens de programação de alto nível

---

- ▶ Bem próximas da linguagem humana
- ▶ O programador pode se concentrar **no que ele quer que o computador faça**, ao invés de ter que detalhar **como o computador executará cada instrução**
- ▶ Abstração
- ▶ Um outro programa (compilador/interpretador) se encarregará de traduzir seu programa para a linguagem de máquina

# Programação

---

“It has often been said that a person does not really understand something until he teaches it to someone else. Actually a person does not really understand something until after teaching it to a computer, i.e., express it as an algorithm.”

(Donald Knuth, in "American Mathematical Monthly," 81)



# Um programa em Java

---

```
public class HelloPrinter {  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
    }  
}
```



# Um programa em Java

---

```
public class HelloPrinter {  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
    }  
}
```

○ que eu quero que seja feito



# Um programa executando a mesma tarefa em Python

---

```
>>> print("Hello, world!")
```

# Um programa executando a mesma tarefa em Python

---

```
>>> print("Hello, world!")
```

- ▶ Digo apenas o que eu quero que seja feito!

# Um comando em Python

---

```
>>> print("3+4+5")
```

# Um comando em Python

---

```
>>> print("3+4+5")
```

```
3+4+5
```

# Um comando em Python

---

```
>>> print(3+4+5)
```

# Um comando em Python

---

```
>>> print(3+4+5)
```

12

# Erros

---

```
>>> printa(3+4+5)
```



# Erros

---

```
>>> printa(3+4+5)
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'printa' is not defined





# Erro?

---

```
>>> print("Helo, wlorld!!")
```



# Erro?

---

```
>>> print(10/0)
```



# Erro?

---

```
>>> print(10/0)
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

ZeroDivisionError: division by zero



# Créditos

---

- ▶ Material feito em conjunto com Aline Paes e Vanessa Braganholo



# Apresentação da Disciplina de Programação de Computadores I



Leonardo Murta

leomurta@ic.uff.br