

# プログラミング演習 (B8)

## 第1回

岡野 訓尚

kokano@fc.ritsumei.ac.jp

# 自己紹介

## ■ 研究テーマ

- システム×制御×情報×人間
- モノを思い通りに動かすにはどんな情報が重要？
- ヒトにとって自然なモノの動きとは？

## ■ プログラミング

- C言語: 学生時代にちょっと最近はArduinoでよく使う
- Java: 会社員時代に
- Matlab, Python: 研究で



**Kunihisa  
Okano**

 Okano Lab

 質問箱

 My old Google site

<https://okanokn.github.io/>

## About

岡野 訓尚 (おかの くにひさ)

[立命館大学 理工学部 電気電子工学科](#)の准教授。博士(工学)。IEEE, 計測自動制御学会, システム制御情報学会, 電子情報通信学会, 電気学会の会員。

Kunihisa Okano is an Associate Professor at the [Department of Electrical and Electronic Engineering, College of Science and Engineering, Ritsumeikan University](#).

## Contact

- 📍 1-1-1 Noji-higashi, Kusatsu, Shiga 525-8577 Japan | 〒525-8577 滋賀県草津市野路東1-1-1
- ✉ kokano at fc.ritsumei.ac.jp

# この講義の目的

## ■ 概要と方法

- C言語は… (中略) … 広範囲に使用されている. 本演習は, 「情報処理」, 「情報処理演習」で学んだC言語のより高度な使用法を演習を通して学びます. また, … (中略) … 「データ構造とアルゴリズム」で学ぶ内容のC言語での実現方法も演習を通して学びます.

## ■ 到達目標

- C言語の文法とプログラミングに関する基本を身に付け, さまざまな問題进行处理, 解決するプログラムを記述することができる.

# 成績評価・注意事項

- 平常点 100%
  - － 出席
  - － 検証課題(2, 3週に1回)
    - ・ 未完成でも提出することを推奨
- 講義中, 検証課題に取り組むうえで, 生成AIの利用は原則禁止
  - － 理由は後述
  - － 調べても, 人に聞いてもわからない場合は使用してよいが, どのように利用したかを再現可能なレベルで明記すること
- 講義室内での飲食禁止

# 今日の目的

- C言語を学ぶモチベーションを上げる
- 演習環境設定
- C言語の復習

# C言語を学ぶモチベーションを上げる

- Q. そもそもプログラミングを学ばなければならないの？
- A. コンピュータを用いた情報処理が今後も拡大するのは確実

# C言語を学ぶモチベーションを上げる

- Q. そもそもプログラミングを学ばなければならないの？
- A. コンピュータを用いた情報処理が今後も拡大するのは確実
- Q. AIがなんでもやってくれるのでは？
- A1. そのAIを作る人材が必要
- A2. (少なくとも現状では)AIの生成したコードは間違いが多い.  
ペアプロの相手としては有効かも

<https://www.gizmodo.jp/2024/05/chatgpt-answers-wrong-programming.html>



# C言語を学ぶモチベーションを上げる

- Q. そもそもプログラミングを学ばなければならないの？
- A. コンピュータを用いた情報処理が今後も拡大するのは確実
  
- Q. AIがなんでもやってくれるのでは？
- A1. そのAIを作る人材が必要
- A2. (少なくとも現状では)AIの生成したコードは間違いが多い.  
ペアプロの相手としては有効かも
- A3. 将来的にAIの誤りが少なくなったとしても, 何らかの方法で人間が品質を担保しなければならない



# C言語を学ぶモチベーションを上げる

- Q. C言語って大昔の言語では？
- A1. 大昔から現在も使われ続けている
- A2. プログラミングの基本を学びやすい(明示的な型付け, ハードウェアとのリンク)

# 今日の目的

- C言語を学ぶモチベーションを上げる
- 演習環境設定
- C言語の復習

# 演習環境

- Google Cloud Shell
  - <https://ide.cloud.google.com/>
  - 無料. Googleアカウント必要
  - 1週間の使用時間上限 50時間
  - 20分間キー入力がないと切断される(ファイルは残る)
  - 12時間経過するとキー入力があっても切断される(ファイルは残る)
  - 120日間使わないとファイルが消える
- ローカル環境が整っている人はそちらでも可
  - Ubuntu > 端末 > emacs &

# プログラミングを学ぶ心構え

- コンピュータの立場になってコードを書こう
  - cf. ラーメンズ『TOWER』より「シャンパンタワーとあやとりとロールケーキ」  
<https://youtu.be/5FejvFO1fM0?t=251>

# プログラムを作成する手順

## 1. 処理手順を検討する

- ノートの上でフローチャートなどを用いて検討する

## 2. プログラムを作成する

- C言語(または他の言語)を用いてプログラムを記述する

## 3. コンパイルする

- 記述したコードをコンピュータが解釈できる形式に変換する

## 4. テスト実行(デバッグ)する

- 実際に動かしてみて, 正しく動くか確認する

# エディタの使い方

- フォルダ一覧ペイン
- エディタペイン
- ターミナルペイン

# よく使うコマンド

- ls: List. このディレクトリのファイル一覧を表示
- pwd: Print working directory. 現在の作業ディレクトリを表示
- cd: Change directory. ディレクトリの移動
- cat: Concatenate. ファイルの中身をテキスト表示
- touch: ファイルの生成
- mkdir: Make directory. ディレクトリの作成
- mv: Move. ファイルの移動
- cp: Copy. ファイルのコピー
- rm: Remove. ファイルの削除

# やってみよう

1. ディレクトリ“01”を作成する
2. “01”直下に hello.c を作成
3. hello.cをエディタで編集して“Hello World!”と出力するプログラムを作成
4. hello.cをコンパイル
  - gcc hello.c
5. hello.cを実行
  - ./a.out
6. hello.cの実行ファイル名をa.outではなくhello.outに指定する



# やってみよう

- 底辺11センチメートル, 高さ3センチメートルの三角形の面積を出力するプログラムを作成して実行する
  - ただし, 底辺, 高さ, 面積はプログラム内部で変数bottom, height, areaとして保持すること

# 型とprintf書式

型宣言	意味	バイト数	範囲（処理系で変わる）	指定子
char	符号付き文字	1	-127~128	%c,%s
int	符号付き整数	4	-32768~32767	%d
long	符号付き長整数	4	-2147483648~ 2147483647	%ld
unsigned char	符号なし文字	2	0~255	%c,%s
unsigned short	符号なし整数	2	0~65535	%hu
unsigned int	符号なし整数	4	0~65535	%u
unsigned long	符号なし長整数	4	0~4294967295	%lu
float	単精度実数	4	$3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$	%f
double	倍精度実数	8	$1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$	%lf