

手続き型プログラミング (実習)

第2回 2020年5月6日

授業説明, 実習の準備

講義情報

講義情報

科目名：手続き型プログラミング基礎

科目コード：CSC.T243

開講時期：1Q

単位数：3 (2-1-0)

担当教員：

伊藤勇太助教, すずかけG3 916 (yuta.ito@c.titech.ac.jp)、

宮藤詩緒助教, 大岡山西9E棟 401 (miyafuji@c.titech.ac.jp)

講義のスケジュール：

講義：（月 7 - 8）， 実習：木 5 - 8

関連講義：

手続き型プログラミング発展（本講義の続き， 2Q開講）

プログラミング創造演習（本講義の発展， 1Q開講）

CSC.T.244 <https://titech-cpp.github.io>

受講上の注意

受講対象者

- a. 情報工学系 2年生
- b. 第5類生、情報理工学院生(入学年度 \geq H30)
- c. 情報工学科/情報工学系/第5類生(入学年度 \leq H29)
- d. 上記以外 (他学科)

受講対象者は原則 **a.** に当てはまる学生

※b, cは受講制限をかける可能性があります.

※dは原則受講不可です.

→b, c (+d) の学生で受講希望者は宮藤に受講希望理由をメールしてください.

H29年以前の学生の単位読み替え

c. 情報工学科/情報工学系/第5類生(入学年度 \leq H29)

の学生は下の表を参考に単位読み替えをお願いします。

旧情報工学科との単位対応

プログラミング第一	情報実験第一	本講義取得単位
単位取得済	単位取得済	履修申告不可
単位取得済	未取得	◎2, 無印1
未取得	単位取得済	○2, 無印1
未取得	未取得	○1, ◎1, 無印1

プログラミング基礎（情報通信系）と同時単位取得は**不可**

プログラミング発展（情報通信系）と同時単位取得は**可**

講義詳細

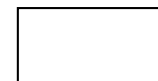
講義スケジュール（仮）

日付	曜日	回数	講義内容
5/4	月曜	1	【講義】 ハードウェア・ソフトウェア、整数型・実数型, 演算子, 標準入出力
5/7	木曜	2	【実習】 手続き型プログラミングへの導入、実習環境の準備
5/11	月曜	3	【講義】 制御構造（条件分岐、ループ）
5/14	木曜	4	【実習】 ハードウェア・ソフトウェア、四則演算と標準入出力に関する実習
5/18	月曜	5	【講義】 関数、再帰、変数のスコープ
5/21	木曜	6	【実習】 制御構造に関する実習
5/25	月曜	7	【講義】 配列、多次元配列、ファイル入出力
5/28	木曜	8	【実習】 関数、再帰に関する実習
6/1	月曜	9	【講義】 文字型、文字列
6/4	木曜	10	【実習】 配列、多次元配列、ファイル入出力に関する実習
6/8	月曜	11	【講義】 探索アルゴリズム・整列アルゴリズム
6/11	木曜	12	【実習】 情報処理課題（基礎）に関するプログラム設計
6/15	月曜	13	【講義】 習熟達成評価
6/18	木曜	14	【実習】 情報処理課題（基礎）に関するプログラム実装・評価



…月曜の講義はオフライン

8



…木曜の講義はオンライン

授業担当者

教員

伊藤 勇太 助教 すずかけG3 916

宮藤 詩緒 助教 大岡山西8 E401

TA (学生アシスタント)

加藤 舜介 (藤井研M2)

武石 健吾 (藤井研M2)

畠山 和久 (徳永研M2)

ドルサ テヨルス (徳永研M2)

津嶋 佑旗 (秋山研M1)

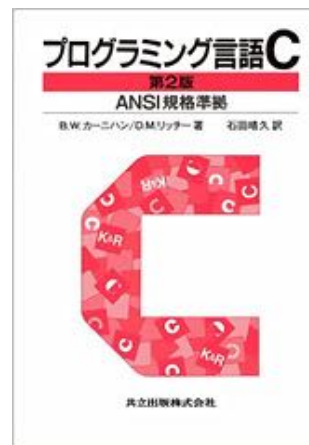
稲垣 雅也 (秋山研M1)



参考書

プログラミング言語C 第2版 ANSI規格準拠

B.W. カーニハン,
D.M. リッチー,
共立出版, 1989



プログラミングに活かす データ構造とアルゴリズム の基礎知識

今泉貴史,
アスキー, 2004



- 講義・実習では用いません.
- あくまで参考書なので購入は必須ではありません.
- 自習に役立ててください.

講義概要・成績評価

講義の概要とねらい

C言語による手続き型プログラミングの基礎的な内容について、講義と実習を通して理論と実践の両面から学びます。

到達目標

1. C言語のデータ型、標準入出力、演算子、制御構造、関数、再帰、配列、ファイル入出力、文字列について理解し、それらを活用したプログラムを作成できる。
2. 複数の探索アルゴリズムや整列アルゴリズムについて理解し、実現するプログラムを作成できる。
3. データ解析等の課題を解決するプログラムを作成し、解決法を説明できる。

成績評価方法

講義部分50%(筆記による期末試験) ???

実習部分50%(毎回の課題+期末レポート課題)

授業の進め方

授業の進め方 1

- 座学講義内容の復習とC言語によるプログラミング実習.
 - 各週, 座学講義1回分ずつ扱う.
 - 授業内の時間で実習課題を提出してもらいます.
 - **実習には毎回参加してください.**
- プログラミング実習
 - 使用する環境: 個人のPC
 - 使用するサイト: AtCoder ※
 - 「C++入門」 <https://atcoder.jp/contests/apg4b>
 - そのほかA問題から講義に合わせて問題を選択していきます.

※AtCoder: 競技プログラミングを開催する国内最大のサイト. オンラインのコンテストだけでなく, 練習用のプログラミング問題を潤沢に用意してくれます.

授業の進め方 2

- AtCoderでの課題提出を行い、TAとの提出確認が終わった時点で授業は終了.
 - 提出確認：**コードの書き方**，**コードの説明**ができるかを確認
 - （例年厳しいらしい…何度も提出しに行くことになるはず）
- 質問，提出について
 - 質問，提出はTA/講師と1対1で行う.
 - 質問，提出は**受付番号表**への記入をもとに順番に対応.
 - **GoogleSheet**受付番号表：<https://bit.ly/2A7CnBz>
 - 質問・提出フロー
 1. **GoogleSheet** 受付番号表に学籍番号・名前等を入れる
 2. **Discord** 質問・提出部屋に
「質問（もしくは提出）があります. 」と投稿
 3. (TA/講師が順番表を確認，学生にTAを割り当てます.)
 4. (提出の場合) TAから提出確認でOKがもらえたら終了.

授業の進め方 3

- 点数の付け方
 - 授業当日**17時**までにプログラム提出
 - 17時までなら再提出可能
 - 17時を過ぎた提出は翌週の授業内で提出確認を受ける
 - 授業時間内に提出確認ができなかった課題は**減点対象**
- SlackとDiscordの住み分け
 - Slack：連絡事項・情報共有
 - Discord：チャット用

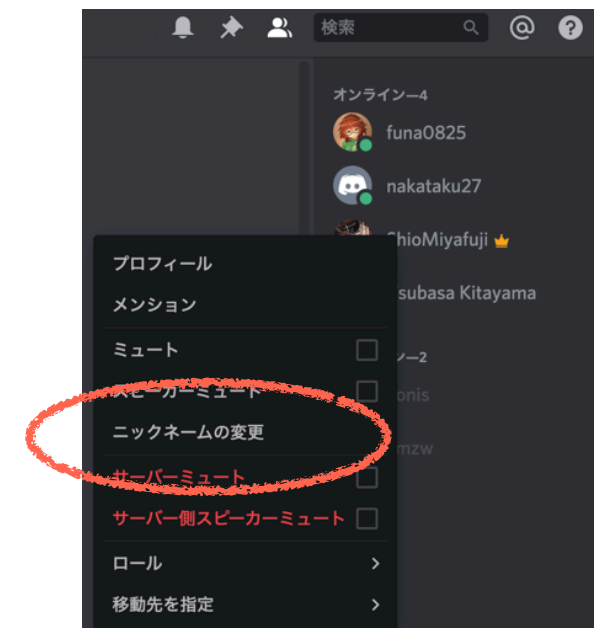
(補足) Discordについて…

- Discord: オンラインゲーム用ボイスチャットアプリ (無料)
 - <https://discord.com>
- Zoom vs Discord

Zoom	VS	Discord
Education課金済み🙏	料金	基本無料🏷️ (容量増は課金)
権限を持つ人間が分割	分割ミーティング	誰でも部屋を作れる (※1)
ホストが ミーティングを立てる	ミーティング開始方法	サーバーに入れば いつでも
バーチャル背景が使える 前回チャットが消える	その他	バーチャル背景なし チャット残る

(補足) Discordの入れ方・設定

- Discord: オンラインゲーム用ボイスチャットアプリ (無料)
 - <https://discord.com>
 - 接続が安定するのでできればアプリをインストールしてください
- 使用アドレスは任意
- 登録時の名前も任意
 - サーバーごとに「ニックネーム」を変更できるので登録名は自由
- サーバー内でのニックネームを変更してください
 - TAは[名前_TA]
 - 学生は[名前_学籍番号]に変更
- ニックネームの変更方法
 - 右枠の自分のアカウントを右クリック→



(補足) Discordのチャンネルについて

チャンネル名	用途
授業中チャンネル (テキスト・ボイス)	授業中のチャンネルです。学生は作業中、基本ここにいてください。
質問・提出部屋 (テキスト)	質問や提出を行う際はこのチャンネルで「質問（提出）があります」と宣言する。
●●さんの部屋 (ボイス)	TA・講師に質問・提出を行うためのチャンネル。指示に従って順番に入ってください。
Staff部屋	TA・講師専用の部屋です。学生ははいることができません。

今日の課題

今日の課題

課題1_1. AtCoderのアカウントを取得しよう！



課題1_2. サンプルコードをコンパイルしてみよう！



課題1_1. AtCoderのアカウントを取得しよう！

- AtCoderのアカウントをつくってください。
 - 既に持っている人はそのまま使ってもOK.
- ログインした状態でプロフィールを表示し，TAに確認をとってください。（←これが本日の提出になります！）



プロフィールの例

課題1_1 補足資料 AtCoderのアカウント取得方法

- AtCoder: 競技プログラミングを開催する国内最大のサイト
 - <https://atcoder.jp>
- 右上の新規登録からアカウントをつくりましょう！
- 登録時のアドレスは任意？
- 登録時の名前も任意？
 - 2Qが終わるまでアカウント名の変更を行わないでください。

課題1_2. サンプルコードをコンパイルしてみよう！

- 各自のコンピュータで環境構築を行い，サンプルコードをコンパイルしてみよう.
- ターミナル上からサンプルコードをコンパイルする様子をTAに見せてください.

```
/*  Kadai1_2
   Sample Code */

#include <stdio.h>

int main(void){
    printf("Hello World!!\n");
    return 0;
}
```

サンプルコード

課題1_2 補足資料 C言語プログラムの作成

- C言語ファイルを新規作成.
 - C言語ファイルの拡張子は「.c」（例：「test.c」）
- ファイル内に先ほどの「サンプルコード」の内容を写経
 - Windowsの場合は “\”（バックスラッシュ）が出ません.
代わりに”¥”（エンマーク）を使ってください.
 - 必ず**半角英数字**で書きましょう！



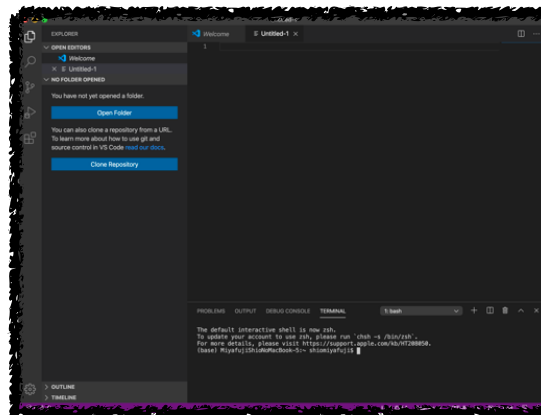
(Mac) 入力文字種類は
右上でチェック

課題1_2 補足資料 ターミナルの起動

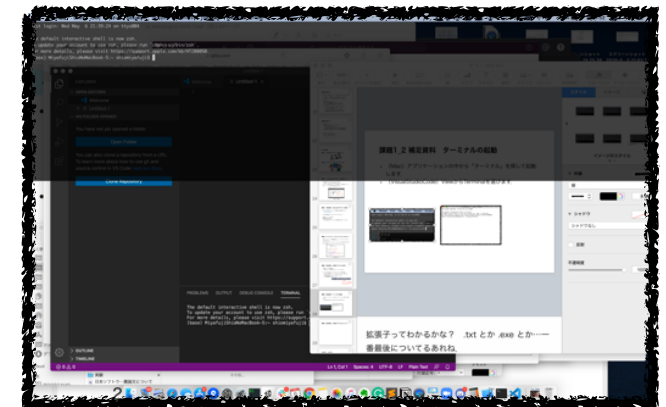
- (Mac) アプリケーションの中から「ターミナル」を探して起動します.
- (VisualStudioCode) ViewからTerminalを選びます.



MacのTerminal



VSCodeのTerminal



MacのiTerm

課題1_2 補足資料 Unixコマンドの基礎

cd (後ろ空欄)	ホームディレクトリへ移動
cd (ディレクトリ名)	指定したディレクトリへ移動
mkdir (ディレクトリ名)	指定の名前で新規ディレクトリ作成
ls	現在ディレクトリ内のファイル一覧表示
mv (今のファイル名) (新ファイル名)	今のファイル名から新ファイル名に変更
rm (ファイル名)	ファイルの削除 (注意：復元できません！！)
rm -r (ディレクトリ名)	指定ディレクトリを消去
<code>rm -rf /</code>	全 部 消 す (絶対にやっちゃダメ)

課題1_2 補足資料 C言語ファイルをコンパイル&実行

- 以下のターミナル上で以下のコマンドを入力

```
i$ gcc sample.c -Wall  
i$ ./a.out
```

コンパイルのコマンド

課題1_2 補足資料 C言語ファイルをコンパイル&実行

- 以下のターミナル上で以下のコマンドを入力

```
i$ gcc sample.c -Wall  
i$ ./a.out
```

実行のコマンド

- Hello World! と出たら

お　わ　り