



情報処理演習II

No. 1

2023.09.26

芝浦工業大学 システム理工学部 機械制御システム学科

担当：桑原

はじめに

「情報処理II」、「情報処理演習II」を受講している皆さんへ

- ・情報処理IIは、各自オンデマンドで学習を進めてください。
中間・期末試験にて成績評価します。
中間試験(10/14)：50点、期末試験(1/16)：50点
- ・情報処理演習IIは、対面形式（5号館1階PC室）で実施します。
火曜1,2限は、情報処理演習IIの説明・演習を実施します。
課題提出にて成績評価します。
前半：50点、後半：50点
- ・不明点・困ったこと等は、教員・TAに相談をしてください。

情報処理演習IIについて

本講義の位置づけ

- 本講義では、C言語を中心に、**データサイエンスのデータ処理としてコンピュータのもつ能力を十分に発揮させ、より柔軟に使いこなすための方法**を習得する。
後半では、力学を中心とした数値解析の方法やプログラミングを学ぶ。
- 資料は基本的に配布テキストにて行います。
講義内容を補完する参考書は必要に応じて各自購入してください。



参考図書：
「C言語によるプログラミング 基礎編 第二版」
内田智史監修, オーム社, ¥2,200円+税

情報処理演習IIについて

学習の進め方

- ・講義前週の土曜日正午以降にScombZからダウンロード・印刷し、予習の上で持参
2 頁/枚でカラー印刷するとメモや書込みがしやすいです。
* 資料は一通り説明しますが、**講義前に必ず一読してください。**
- ・講義は、資料の説明→C言語プログラミング課題演習→解説(次回)の順番で進めます。
特に、演習では、必ず、自分でコーディング（プログラム実装）しましょう。
手を動かして早く慣れた人ほど習熟度が高くなります。
- ・課題演習の解答サンプルプログラムを理解し、復習をしてください。
サンプルプログラムをただ眺める・写経するだけでは何も身に付きません。
各コードの意図・やっていることを他人に説明できて初めて「理解した」と言えます。
- ・予復習を怠る、遅刻する、授業中にネットサーフィンしていると、
すぐに理解が追い付かなくなります。計画的に学習・行動するように心がけましょう。

課題について

課題の提出要領

- ・提出先：GoogleDriveにある「課題提出¥課題No.」に提出してください。
* ドラッグ&ドロップではなくコピー&ペーストで提出してください（事故防止）。
- ・作成したファイルは、“bq22000_0926_work_11.c”のように、“学籍番号_出題月日(4桁)_work課題番号.拡張子”としてください。
- ・原則、**出題時限の終了時まで**に提出（それ以降の提出は再提出扱いとする。）
- ・再提出の場合には、「bq22000_0926_work_11_r1.c」のように末尾に「r」を追加する
→ 提出期限内で最も新しいものが評価対象

以上が守られていないものは採点されないので要注意

C言語とは？

C言語はUNIX(OSの1種)を記述するためのプログラミング言語として誕生（1970年代）

- ・プログラミング言語＝「コンピュータに仕事をさせるために作業手順を指示する、特別な言語」。
- ・プログラミング言語で書かれた文章：**プログラム（コード）**
プログラムを書くこと：**プログラミング（コーディング）**

現在は、汎用プログラミング言語として、多くの目的に使用。

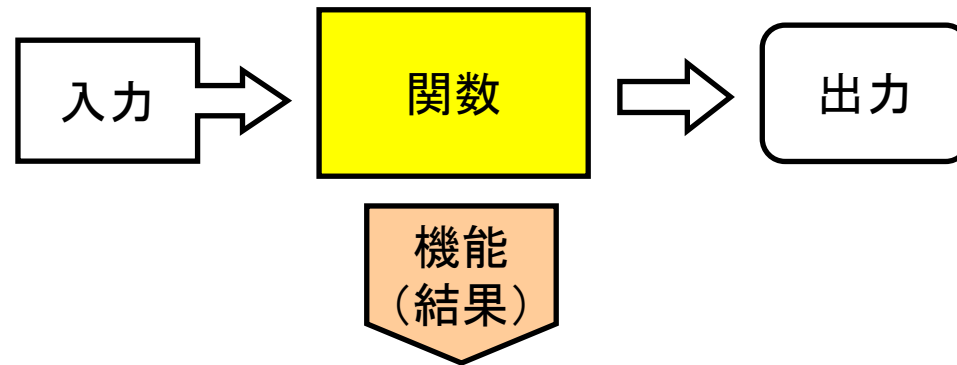
- ・**H/Wとの親和性が高く**、ロボットや機械のリアルタイム制御ではデファクトスタンダード
 - ・C++、Objective-C などは上位互換（オブジェクト指向）、Java にも影響を与えている。
- C 言語は現存するほぼ全てのプログラミング言語の基礎となっている。

C言語の特徴

高級言語の一種だが、メモリに直接アクセスすることも可能（後に学ぶ"ポインタ"）

プログラムは関数の集合からなる。

- ープログラムの「関数」はある機能のまとまりを意味する
- ー数学の関数と同様に、変数（入力）と戻り値（出力）がある。
- ーよく使う関数はライブラリとして提供されている。
- ー定義されている関数もある（main関数：一番最初に呼び出される関数）。



C言語の仕組み

C言語は「コンパイラ言語」の一種

プログラムを書くこと
= **コーディング**



```
#include <stdio.h>

main()
{
    int n;

    for (n = 0 ; n < 10 ; n++){
        printf("Hello World!¥n");
    }
}
```

C言語で書かれたファイル（プログラム）
= **ソースファイル**
（ソースコード、ソースプログラム、略してソース）

書き換え作業 = **コンパイル**

コンピュータが理解できる言語（機械語）で
書かれたファイル（プログラム）

= **実行ファイル**（バイナリコード）

```
01011101001001010100101011
01001001001010010101010010
10010101010010101110100101
011101010111010101010100
10111011010110100101000111
11101100101001010101010010
11010101010010101001010101
01010100101010101001010100
10101010100101010010101001
01001010100101010101010010
1010
```



プログラミングの流れ

1. C言語の文法に従ってソースコードを書く（コーディング）

「書く」ためのソフトウェア = **エディタ**

1. ファイル名の**末尾は「.c」**とする。

例：test.c （この「.c」は**拡張子**と呼ばれる。）

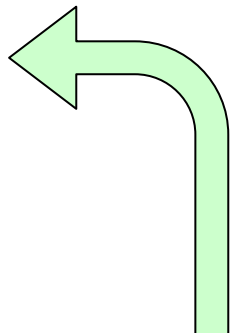
2. ソースコードをコンパイルする

1. コンパイルする（実行ファイルを生成する）ためのソフトウェア = **コンパイラ**
2. 実行ファイルは「.exe」
3. 文法が間違っているとエラーが返って来る。

3. 実行ファイルを実行する（テスト）

コンパイル時にエラーがなくても、正しい実行結果が得られなかったり、実行できなかったりすることもある。

* プログラムの誤りや欠陥 = **バグ**



1.に戻ってソースコードを
修正し、2.および3.のステップ
を繰り返す作業

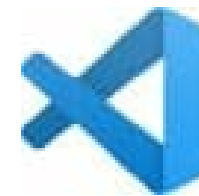
= **デバッグ**



本講義でのプログラミング環境

C言語のプログラミング環境

- ・エディタ：ソースを編集するもの 例：メモ帳、秀丸など
- ・コンパイラ：ソースから実行ファイルを生成するもの 例：MinGW
- ・統合環境：エディタ・コンパイラ・デバッグツールなどをまとめたもの 例：VSCode



[Visual Studio Code](https://code.visualstudio.com/)

本講義では、PC室のPCを使うことを考慮し、以下の環境を前提とします。

ただし、自己責任の範囲で自分のPCに統合開発環境をインストール・使用してもOKです。

- ・エディタ：メモ帳、秀丸、EmEditor (フリー版)など任意。
- ・コンパイラ：gcc (Cygwinインストールオプションとして入手可)
 - * Cygwin：Windows OS上にUNIXライクな環境を提供する互換ソフト
 - * 状況に応じてですが、**課題ではプログラム出力を画面キャプチャしたものを提出して頂くこともあります。**



Cygwinを利用したソースファイルのコンパイル・実行方法

0. 作業用フォルダを作成 : 「C:¥Project」に「test」というフォルダを作成
* 必要に応じて読み替えてください。
1. エディタでソースコードを作成 : test1.cとする。作業用フォルダ内に保存する。
2. Cygwinを起動し、作業用フォルダに移動する : cdコマンドを使用
\$ cd C:¥Project¥test
3. コンパイル : gccコマンドを使用
\$ gcc -o test1 test1.c
4. 生成された実行ファイル(.exeファイル)を確認 : lsコマンドを使用
\$ ls
5. 実行
\$./test1

C言語のソースをコンパイル・実行してみる

- **赤字は毎回必要**です。**コードは半角文字で書いてください（空白も半角にすること）**
- 「/*」と「*/」で囲まれる部分はコンパイル時に無視される（コメントの記入に便利）。
- #includeは外部ファイルの読み込みを意味する。この場合、PCの特定の場所にある「stdio.h」というファイルの内容を読み込む。拡張子が「.h」のファイルはヘッダファイルと呼ばれている。
- main()の後の「{」から「}」までが、このプログラムの本体となる。この部分を**メインルーチン**と呼ぶ。
- main()の部分は別の書き方をすることもある。
- printfは（ ）内の「」で囲まれた文字列を画面に表示する**関数**。ただし、「¥n」は「改行」を意味する。

サンプルプログラム 1

```
/*  
 * サンプルプログラム 1  
 */  
#include <stdio.h>  
  
void main()  
{  
    printf("Hello, world! ¥n");  
}
```

実行結果 1

```
Hello, world!
```

コーディング作法の基本

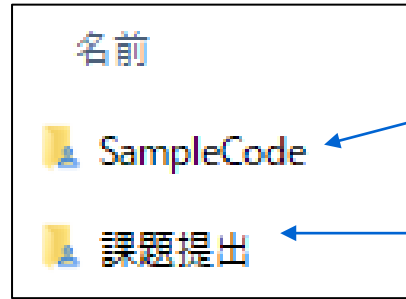
- 先頭にプログラムの説明をコメントとして書く。
→ 他人が読んで分かりやすいように必ずコメントを入れる（時間が経つと作成者も思い出せない）
- #include文の後は空行を入れる。
- main()の後に改行を入れ、「{」は次の行に書く。メインルーチン終わりの「}」の直前にも改行を入れる。
- **メインルーチンは先頭にタブ（Tabキー）を打って行頭から一定スペースを空ける。**
- 行末は「;」（セミコロン）で終わる。
- 最後は改行で終わる（これを忘れるとエラーが出る）。

サンプルプログラム 1

```
/*  
 *サンプルプログラム 1  
 */  
#include <stdio.h> 改行  
空行  
void main() 改行  
{ 改行  
    タブ printf("Hello, world! ¥n"); 改行  
} 改行
```

PCログイン後の操作

- Google Driveで共有フォルダ「情報処理演習II」にアクセス



サンプルコードなどをここからダウンロードする

課題提出用

- 「SampleCode」フォルダから“sample1.c”をコピーして、自分のPCの任意の場所にダウンロードする。

- Cygwinウィンドウを起動し、ファイルのあるフォルダ（ディレクトリ）まで移動する。

例. 「C:¥Users¥Desktop」に移動する場合のコマンド：

```
cd /cygdrive/C/Users/Desktop
```

- コンパイル

```
gcc -o sample1 sample1.c
```

- 実行

```
./sample1
```