

計算機プログラミング

Computer Programming

第1回目

April 15, 2021

(10:30 ~ 12:00)



担当教員: Thi Thi Zin (ティティズイン)
<thithi@cc.miyazaki-u.ac.jp>



自己紹介

- 担当教員: **Thi Thi Zin** (ティティズイン)
- 職名: 教授
- 所属: 工学教育研究部
- 専門分野: 画像処理・認識・理解、マルチメディア検索、
姿勢・動作の認識

授業中、いつでも気軽に質問を

- 何回でも OK
- メールアドレス: thithi@cc.miyazaki-u.ac.jp
- 研究室 : E棟703号室



教育目的

- 本講義では、
C言語を初歩から学び、**簡単なプログラムができる**ことを最初の目的とする。
次に、**計算機ハードウェアを動かすためのソフトウェアの概念**を理解できるようにする。
- さらに、他のプログラミング言語に大きな影響を与えている**C言語**を用いて、
やさしい課題を数多くこなしながらプログラミングの基礎の習得を目指す。
- また、**プログラミングを通じて、問題分析、課題解決のための知識活用、物事の論理的思考を学ぶ。**

(キーワード) C言語、変数、演算子、配列、文字列処理、関数、
ファイル処理、ポインタ



教育目標

- Visual Studio環境でC言語プログラムをコンパイル、リンク、実行できる。
- C言語の理論記述を理解する。
- C言語のプログラムが読める、プログラムを作成することができる。
- 計算機における情報表現を理解する。
- プログラムが何をしているのかを理解する。




授業計画

連絡事項

- 第1回目(4月15日)・第2回目(4月22日)遠隔授業(オンデマンド型)で行う
- 第3回目から対面で行う予定であるが、宮崎県における新型コロナウイルス感染状況によって、変更する可能性が考えられる
- 授業計画など WebClassより随時、連絡する

第1回目:4月15日	Visual Studio 説明と基本操作
第2回目:4月22日	プログラミング入門と実行手順
第3回目:4月30日	printf() と変数
第4回目:5月 6日	scanf() と演算
第5回目:5月13日	if と演算子
第6回目:5月20日	switch, for その1
第7回目:5月27日	for その2, while
第8回目:6月 3日	中間試験
第9回目:6月10日	配列
第10回目:6月17日	文字列と文字列処理
第11回目:6月24日	関数
第12回目:7月 1日	ファイル処理 (1)号
第13回目:7月 8日	ファイル処理 (2)
第14回目:7月15日	ポインタ
第15回目:7月29日	期末試験
第16回目:8月 5日	答案返却と解説



第1回目(4月15日)・第2回目(4月22日)は WebClassより実施

遠隔授業を受けるルール

- 課題の内容、小テスト、レポートの情報は評価に関わるので、他人に提供しないこと
- 遠隔授業の内容を勝手にSNS等で大学外第3者が観覧できるようにアップロードはしないでください。違法行為に当たる場合があります。

出席条件

- Webclassへのログインがあり、準備したテスト、アンケート、レポート等の提出があることで出席とする
- 締め切りまでに問題の解答をレポートとして提出する



文献・教材

- 教科書：計算機プログラミング演習で用いる教科書と同じです
「苦しんで覚えるC言語」
- 参考書1：高橋麻奈「やさしい C」第4版、
ソフトバンククリエイティブ、
ISBN978-4-7974-7098-0

教科書は「計算機プログラミング演習」で購入していると思うので、この授業では適宜、資料等を配布する。

- 本講義の前半9週目までは、プログラミング演習と連携して行う予定である。
- 講義の理解のために、随時、レポートを課して、自己学習能力の育成を図る。
- 中間・期末試験だけではなく、小テストを実施し、学習内容の理解を深めると同時に達成度を計る。



本日の内容

- コンピュータプログラミング
- Hello world !から(コンパイルと実行)
- ソースコードの書き方



コンピュータプログラミング

- プログラム

計算機に実行させる処理手順をある一定の記述言語で具体的に記した表現。

- プログラミング

計算機のプログラムを作成すること

- コンパイラ

高級言語で書かれたプログラムを機械語に一括して翻訳するプログラム

- 高級言語(高水準言語)

人間が使う言語に近い要素に基づくプログラミング言語の総称



高級言語

- **FORTAN (1957)** 数値計算, 技術計算
- **LISP (1959)** 記号処理, 人工知能
- **COBOL (1960)** 事務処理
- **BASIC (1965)** 汎用, 初心者向け
- **Pascal (1971)** プログラミング教育用
- **C (1971)** システム記述, 汎用
- **Prolog (1973)** 論理プログラミング, 人工知能, 自然言語の構文解析
- **C++ (1985)** オブジェクト指向プログラミングが容易になるようにCを拡張
- **Java (1996)** C言語を基盤とした構文; オブジェクト指向; 汎用

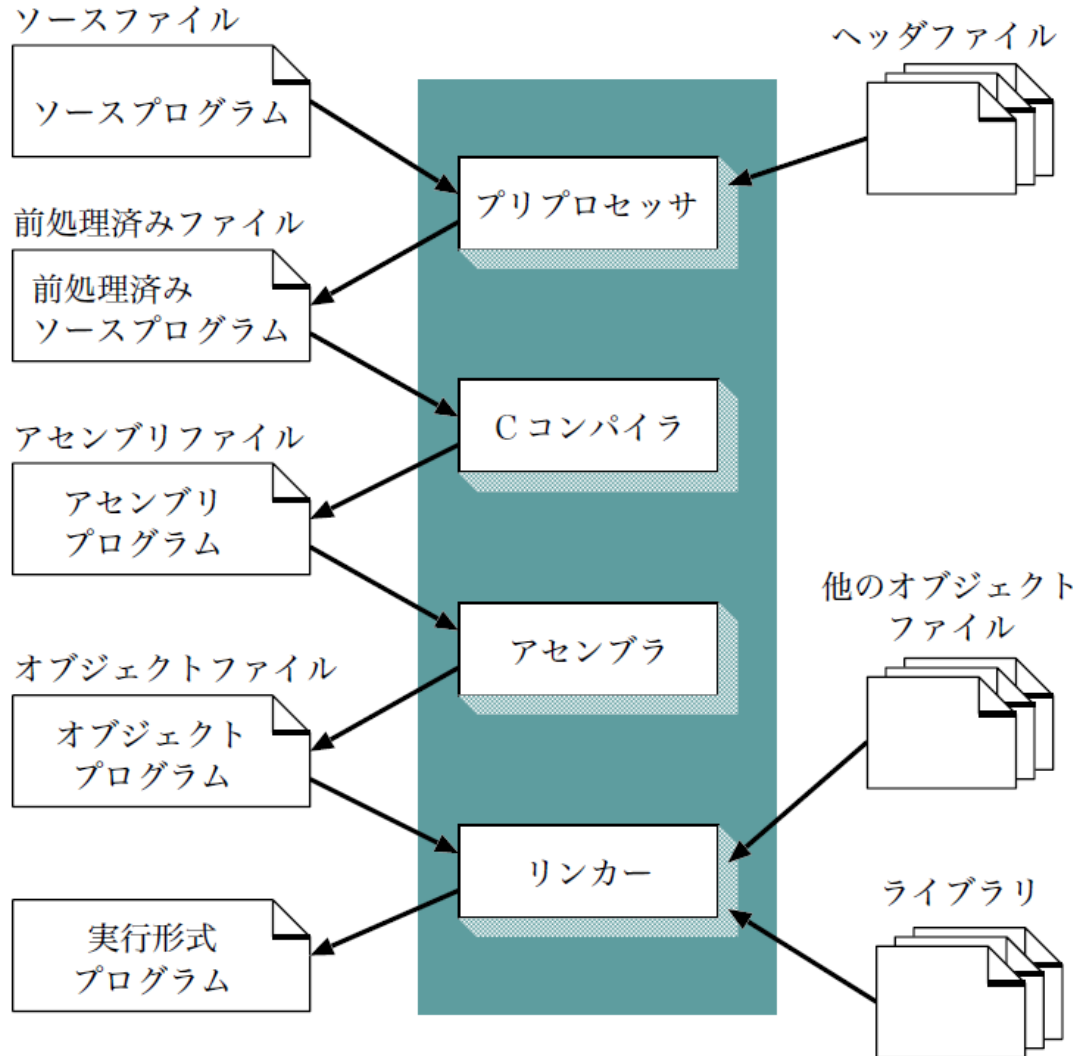
この他にもたくさんのプログラミング言語があります。



機械語とアセンブリ言語

- **機械(マシン)語**
コンピュータが直接理解できる「ことば」
0 と1 の並びでできている
- **アセンブリ言語**
プログラミング言語の一種
アセンブリ言語の命令と機械語の命令はほぼ一対一の
対応をもつ
- **アセンブラ**
アセンブリ言語から機械語への翻訳プログラム

C コンパイラ



人間に分かりやすく複雑な機能や構文を持つ高水準プログラミング言語(高級言語)で書かれた**コンピュータプログラム(ソースプログラム)**を、**コンピュータが解釈・実行**できる形式に一括して変換するプロセス

Hello world ! と表示してみる

画面(コマンドライン)にHello world!と表示する

sample1.c

```
1  /*
2      2021.04.15
3  */
4  #include<stdio.h>
5  int main(void)
6  {
7      printf(" Hello world! \n");
8      return 0;
9  }
```

ソースコードの書き方

sample1.c

```

1  /*
2      2021.04.15
3  */
4  #include<stdio.h>
5  int main(void)
6  {
7      printf(" Hello world! ¥n");
8      return 0;
9  }
```

プログラムの実行には影響を与えず、プログラムの説明を書き加えておきたい場合、C言語では、
 /* コメント文 */ のように書く。

- ※ 半角英数字で入力
- ※ 空白も半角スペース
- ※ 字下げには [Tab] キーの使用（2、7、8行目）

sample1.c

```

1  /*
2      2021.04.15
3  */
4  #include<stdio.h>
5  int main(void)
6  {
7      printf(" Hello world! ¥n");
8      return 0;
9  }
```

#include命令

#include<>
c言語にはすでに多くの作業用
命令(サブルーチン)が用意さ
れている。

画面への出力命令(**printf**)や、キーボードからの入力命令(**scanf**)も、コンパイラが用意しているサブルーチンであり、これを使うときには、サブルーチンの定義情報を知らなければならない。


#include命令は、自分のプログラムの先頭に、サブルーチンに関する情報ファイルである**ヘッダーファイル**(この場合stdio. h)を接続する命令である。

画面(コマンドライン)にHello world!と表示する

```
/*  
    2021.04.15  
*/  
#include<stdio.h>  
int main(void)  
{  
    printf(" Hello world! ¥n");  
    return 0;  
}
```

printf を使うために必要

main 関数を終了



最初に

`int main(void){`

と書かれた次の文から実行される！

“ ” で囲まれた内容
を画面に表示する

➤ プログラムには「main」関数が一つ必要

mainはC言語の予約語であり、プログラムの実行開始部分（メインルーチン）を示している

プログラムは main 関数から実行される

➤ 上から順に実行されてreturn文で終了する

続いて、「hello, 」と「 world! 」が別々の行で表示されるように sample1 のプログラムを変更してみましょう。

```

1  /*
2      2021.04.15
3  */
4  #include<stdio.h>
5  int main(void)
6  {
7      printf("hello, ¥n world! ¥n");
8      return 0;
9  }
```

セミicolon;
で命令を完結

¥n は改行を表す

以下のように2つに分けても同様の結果が得られます。
sample1 のプログラムを変更してみましょう。

sample1.c

```

1  /*
2      2021.04.15
3  */
4  #include<stdio.h>
5  int main(void)
6  {
7      printf(" hello, ¥n");
8      printf(" world! ¥n");
9      return 0;
10 }
```

主な記号の読み方

#	シャープ
"	ダブルクォーテーション(二重引用符)
,	カンマ
.	ピリオド
:	コロン
;	セミコロン
@	アットマーク
*	アスタリスク
/	スラッシュ
&	アンパサンド
!	エクスクラメーションマーク(感嘆符)

引数

printf (“1足す1は%dです” , 1 + 1)

文字列

式

printf …… 出力を行う関数

%d …… 変換指定

プロジェクト名 : **sample2** ソースファイル名 : **sample2.c**

```

1  /*
2      2021.04.15
3  */
4  #include <stdio.h>
5  int main(void)
6  {
7      printf("1足す1は%dです ¥n", 1+1);
8      return 0;
9  }
```

1足す1は2です。

- ※ 半角英数字で入力
- ※ 空白も**半角スペース**
- ※ 字下げには **[Tab]** キーの使用

sample2.c を次のように変更

```

1  /*
2      2021.04.15
3  */
4  #include <stdio.h>
5  int main(void)
6  {
7      printf("%dは%dの2倍です¥n", 2,1);
8      return 0;
9  }
```

%d は複数使うことも可能

整数以外の値を埋め込みたい場合...

変換指定	
%d	整数(10進数)
%f	小数(浮動小数点数)
%s	文字列

加算(足し算)以外の演算子は...

数学	C言語
+ (加算)	+
- (減算)	-
× (乗算)	*
÷ (除算)	/
... (剰余算)	%

sample2.c を次のように変更

```

1  /*
2      2021.04.15
3  */
4  #include <stdio.h>
5  int main(void)
6  {
7      printf("浮動小数点数 %f ¥n ", 0.123456);
8      return 0;
9  }
```

sample2.c を次のように**変更**

表示される桁数を指定するには...

```

1  /*
2      2021.04.15
3  */
4  #include <stdio.h>
5  int main(void)
6  {
7      printf("浮動小数点数 %.2 f ¥n ", 0.123456);
8      return 0;
9  }
```

sample2.c を次のように変更

```

1  /*
2      2021.04.15
3  */
4  #include <stdio.h>
5  int main(void)
6  {
7      printf(" 文字列 %s ¥n ", "abc");
8      return 0;
9  }
```

sample2.c を次のように**変更**

表示する最大文字列長を指定するには...

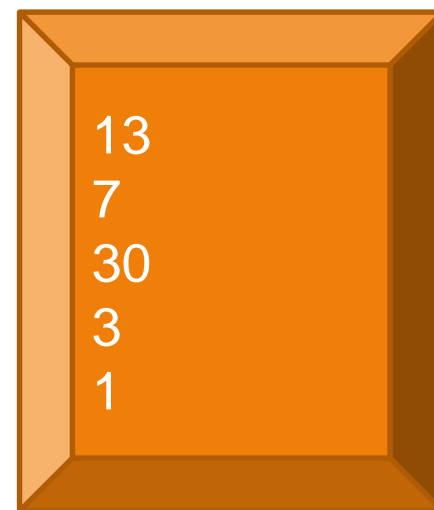
```

1  /*
2      2021.04.15
3  */
4  #include <stdio.h>
5  int main(void)
6  {
7      printf(" 文字列 %.2s ¥n ", "abc");
8      return 0;
9  }
```

プロジェクト名 : `sample3` ソースファイル名 : `sample3.c`

```

1  #include <stdio.h>
2  int main(void)
3  {
4      printf("%d¥n", 10+3);
5      printf("%d¥n", 10-3);
6      printf("%d¥n", 10*3);
7      printf("%d¥n", 10/3);
8      printf("%d¥n", 10%3);
9      return 0;
10 }
```



本日のまとめ

- 第1回目の内容について、理解度を図るため、次回から簡単なプログラムを書いて、レポートを提出してもらう予定である。
 - 本資料の内容について理解を深めておくこと