

Cプログラミング1 (再履修者対象) 第1回

講義では、Cプログラミングの基本を学び

演習では、やや実践的なプログラミングを通して学ぶ

成績評価方法

- シラバスに準ずる, ただし、、、
 - 演習O→試験X : 演習も不合格。来年も宜しく。。。
※演習が受かって試験に落ちるはずがない
 - 演習X→試験X : 来年も宜しく。。。
- 付帯事項
 - 出席回数
 - 講義回数の2 / 3以上を満たした場合のみ,
期末試験の受験資格あり
 - 複製時のペナルティ
不合格の判定

Cプログラミング言語

- これは『言語』である

ただし、日本語や英語は**自然言語**

人間のコミュニケーションの手段として**自然発生的**に生まれた

Cプログラミング言語は **人工言語**

ある**目的**のために**開発**され、**文法**が定義されている

目的：コンピュータを効率的かつ正確に制御する

開発：1972年にAT&Tベル研究所のデニス・リッチーらによる

文法：させたい事柄によって書き方が決まっている

人間がコンピュータにさせたいことを伝える手段

コンピュータ

注: PC だけではない ---- PC: Personal Computer

小はArduino, Raspberry Piなどから大はスーパーコンピュータまで

コンピュータの得意なこと：記憶する、同じことを繰り返す（飽きない）、大量のデータの処理（めげない）、つながる（ネットワークを介して）

不得意なこと：空気を読む、融通をきかせる、誤りを許す、曖昧な命令を扱う、...

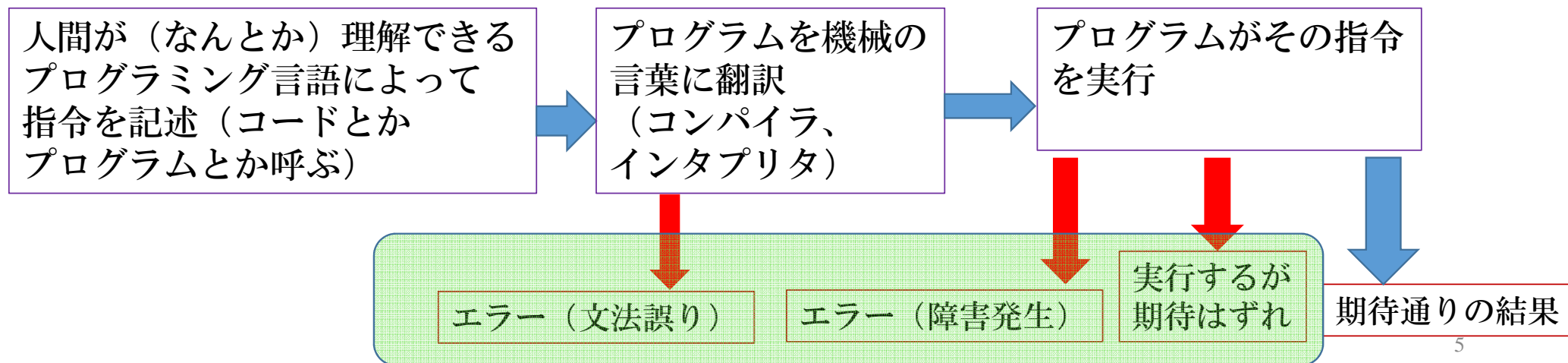
要するに：**タフだけど馬鹿である**

したがって、賢い人間が、馬鹿なコンピュータに分かるように指示しなければならない
= **文法に沿った、正しく動くプログラムをコンピュータに与える**

Cプログラミング言語 (続)

Cプログラミング言語は、人間からのコンピュータの指令の手段
ただ現状では（まだ）人間が音声でコンピュータにいろいろな指令をすることはできていない(簡単な操作は可能であるが)

そこでCプログラミング言語が想定する流れ：



プログラム作成

大きく分けて5種類の作業

1. 問題を**分析**し、どのような入力に対しどのような出力をするかを明確にする。そして解き方（**アルゴリズム**）の概略を考える(**要求分析**、**概要設計**)
2. 1で考えたアルゴリズムに基づき、必要な**データ構造**と作業手順（**フローチャート**など）を書く(**詳細設計**、**プログラム設計**)
3. 設計図に基づき**プログラム**として書く（**コーディング**）
4. プログラムを**コンパイル**、**実行**する（**プログラムテスト**）

エラーが起きた時にはどのような種類のエラーかを分析し、誤りを訂正（**デバッグ**という）して、望む結果が得られるまで繰り返す

5. 幾つかのテストを繰り返した後に、取説（取扱説明書）や報告書を書く

プログラム作成とCプログラミング言語1の関係

大きく分けて5種類の作業

国語力を磨くこと

1. 問題を**分析**し、どのような入力に対しどのような出力をするかを明確にする。そして解き方（**アルゴリズム**） **基本的なアルゴリズムを学ぶ**
2. 1で考えたアルゴリズムに基づき、必要な**フローチャート**など）を書く（**詳細設計、プログラム設計**） **配列、繰り返し、関数を学ぶ**
3. 設計図に基づき**プログラム**として書く（**コーディング**） **言語表現力を磨く**
4. プログラムを**コンパイル、実行**する（**プログラムテスト**） **実践し体験する**
エラーが起きた時にはどのような種類のエラーかを分析し、誤りを訂正（**デバッグ**という）して、望む結果が得られるまで繰り返す
5. 幾つかのテストを繰り返した後に、取説（取扱説明書）や報告書を書く
レポートとして書く。実践あるのみ

Cプログラミング言語の主要な言葉

```
#include
#define
main
void
int    float    double    char
printf scanf
for    while        do
if     else
return
exit
switch    case
h
stdio    stdlib    math    string
fopen    fclose
```

これらは
英語の単語
もしくは
英語の単語の省略形
もしくは
上2つの混合形

だから
基本的な英語の単語力が必要

Cプログラミング言語の主要な言葉

#include
#define
main
void
int float double long char
printf scanf
for while do
if else
return
exit
switch case
h
stdio stdlib math string
fopen fclose

茶色で表示されたものは英語の単語
わからないものがあったら調べてみよう

緑色は単語の一部

int	integer
char	character
math	mathematics
h	header

その他は単語の寄せ集め

printf = print formatted
scanf = scan formatted
stdio = standard input output
stdlib = standard library
fopen = file open
fclose = file close

最初のCプログラム: Hello, world!

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("Hello, world!¥n");
```

```
    return 0;
```

```
}
```

Cの処理の流れ

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("Hello, world!¥n");
```

```
    return 0;
```

```
}
```

コンパイラ (ここでは bcc32を仮定)

文法チェック

関数や変数のチェック

(知らない言葉が使われてないか)

機械語ファイルを出力

⇒ Windows OSならexe

ファイルができる

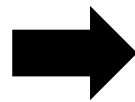
プログラムの実行

WindowsOSの場合、exeファイル名をコマンドプロンプトに入力⇒実行される

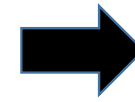
Cにおける関数：入力、処理、出力



入力



処理



出力

プログラム(関数の呼び出し)

$n = \text{func}(a, b, c)$

出力

処理

入力

最初のCプログラム: Hello, world!

include: 含む --- 次のファイルを『取り込め』

```
#include <stdio.h>
```

stdio : 標準の入出力を可能にする
h : ヘッダファイル

```
int main(void)
```

main: 主要な --- Cのプログラムで必要不可欠な関数

```
{
```

void : 空虚 --- 引数をとらない

```
    printf("Hello, world!¥n");
```

printf: 体裁(format)を整えて出力表示(print)

```
    return 0;
```

```
}
```

return: 返す --- main関数は0を値として返す
0: 「正常に実行された」ことを表す

質問

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("Hello, world!¥n");
```

```
    return 0;
```

```
}
```

このmain関数は

何が入力

どのような処理をする

何が出力

だろうか？

注意： 『関数の出力』は「返り値」
や「戻り値」と呼ばれる

出力関数と混同しないように！

Cプログラムの注意点

```
#include <stdio.h>

int main(void)
{
    printf("Hello, world!¥n");

    return 0;
}
```

- (1) ファイル名を hello.c のようにする
つまり **ファイル拡張子** を c とする
- (2) **全角文字** を使ってはいけない
(例外：文字列の中はよい)
特に、全角の空白文字には注意
- (3) #include文では ファイル名を **<と>**
もしくは **"と"** でくる。つまり
stdio.h はPCにあるファイル！
- (4) **単語の中** にスペース（空白）
をいれてはいけない。
単語間 にはスペースをいれること
- (5) 関数mainは必ず一つ必要：これが
プログラムを起動した時に真っ先に
呼び出される
- (6) 丸括弧と波括弧の使い分け
- (7) 実行文の最後には ; が必要

それではやってみよう

```
#include <stdio.h>

int main(void)
{
    printf("Hello, world!¥n");

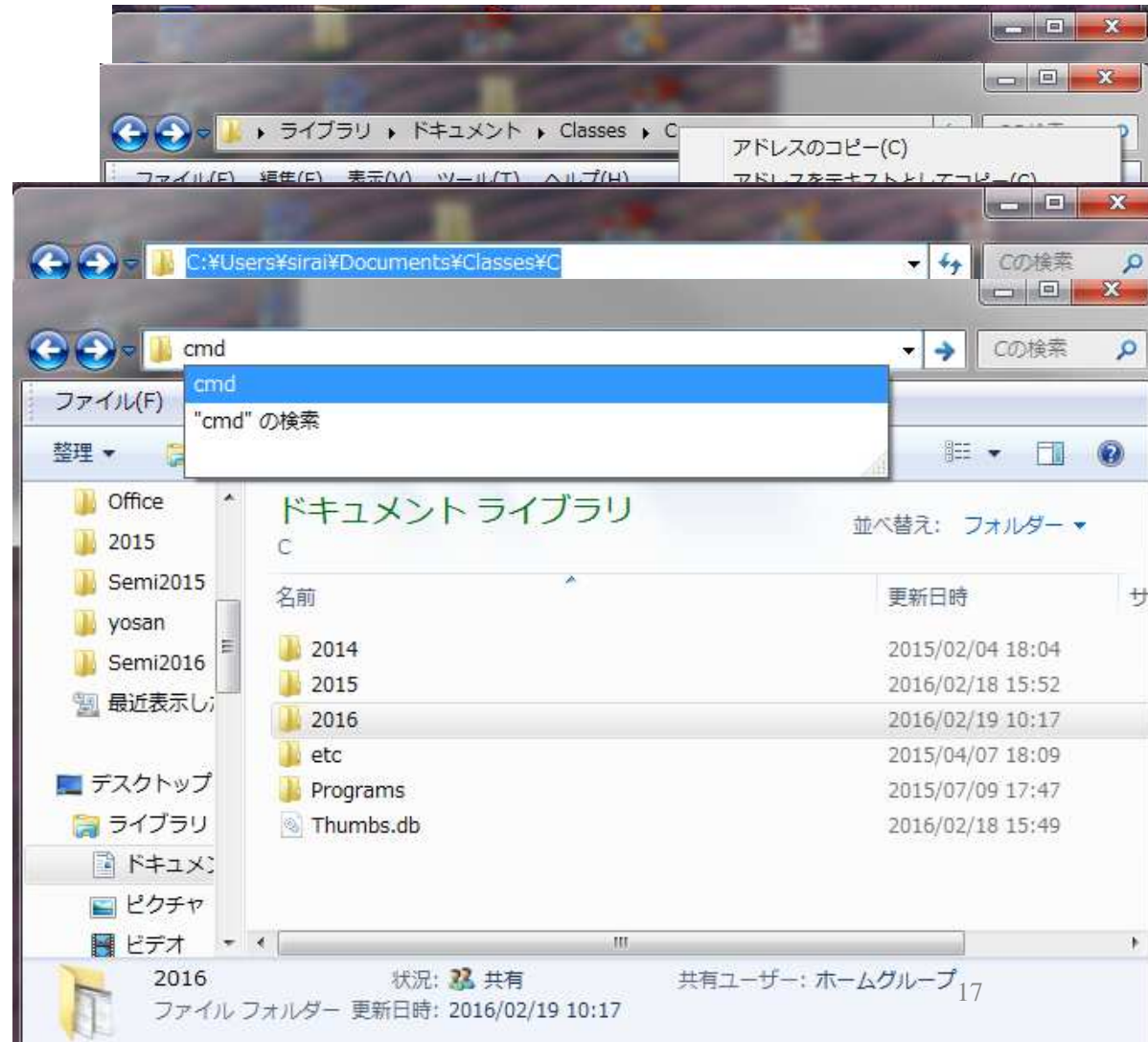
    return 0;
}
```

これを秀丸エディタやTeraPadを使って
打ち込む
絶対に Word や メモ帳、ワードパッド
を使ってはいけない

入力が終わったら、Hドライブに
hello.cとして保存
Hドライブでコマンドプロンプトを開く
bcc32 hello.c
を実行し、デバッグの後
hello.exe を入力して実行させる

Hドライブでコマンドプロンプトを開く

1. Hドライブを表示させる
(白井のフォルダーを例にする)
2. ファイルパスにマウスをもっていき、右クリック
3. 『アドレスの編集』を左クリック
4. 青地のラインにcmdと入力しエンターキーを押す



それではやってみよう

```
#include <stdio.h>

int main(void)
{
    printf("Hello, world!¥n");

    return 0;
}
```

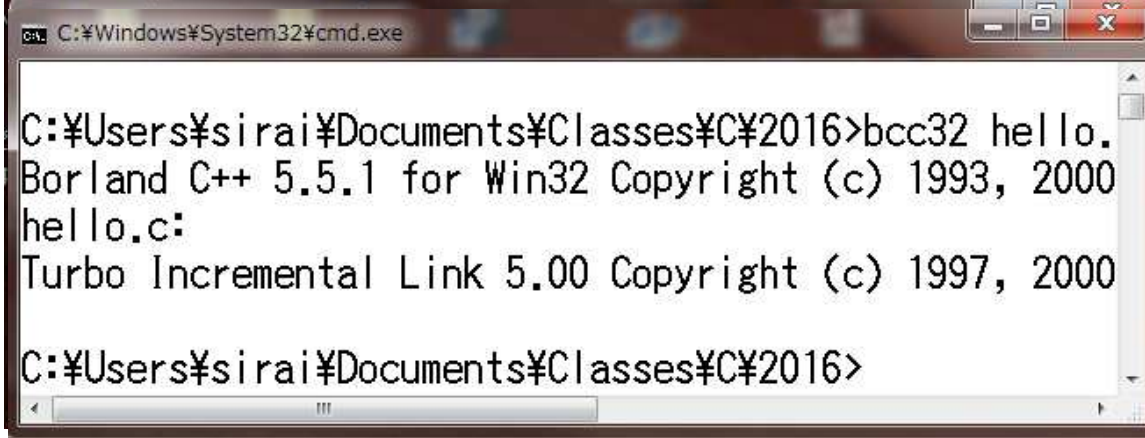
これを秀丸エディタやTeraPadを使って
打ち込む
絶対に Word や メモ帳、ワードパッド
を使ってはいけない

入力が終わったら、Hドライブに
hello.cとして保存
Hドライブでコマンドプロンプトを開く
bcc32 hello.c
を実行し、デバッグの後
hello.exe を入力して実行させる

コマンドプロンプトにおいて

1. bcc32 hello.c

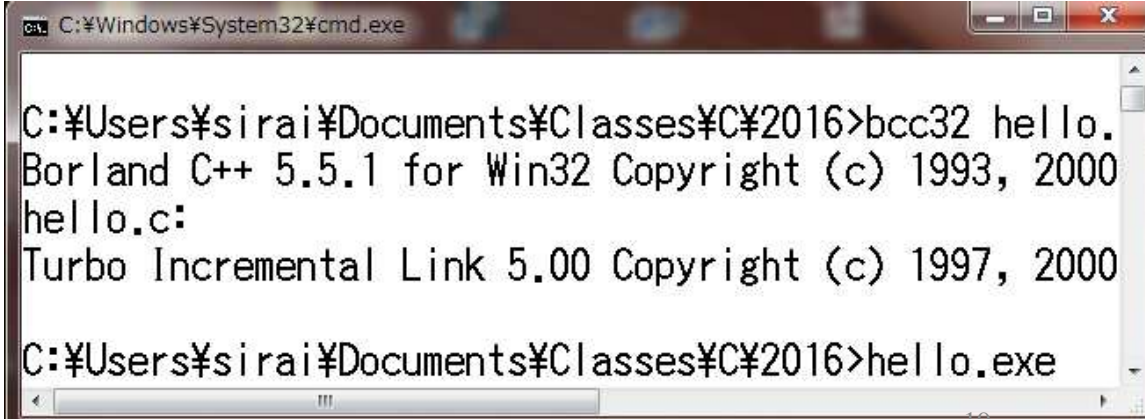
と入力(エンターキーを押す)



```
C:\Windows\System32\cmd.exe
C:\Users\sirai\Documents\Classes\C\2016>bcc32 hello.
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000
hello.c:
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000
C:\Users\sirai\Documents\Classes\C\2016>
```

2. hello.exe ができていたら

hello.exe と入力(エンターキーを押す)



```
C:\Windows\System32\cmd.exe
C:\Users\sirai\Documents\Classes\C\2016>bcc32 hello.
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000
hello.c:
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000
C:\Users\sirai\Documents\Classes\C\2016>hello.exe
```

最初の課題

ウェブページにある numberGame.c のプログラムをなるべく忠実に「自分で」打ち込み、コンパイルし、実行してみよ

エラーが出た場合は、その理由を考えて、修正し、ゲームができるようにすること

最小の手間で数を当てられる方法を考えよ。

提出物: 打ち込んだコード、打ち込みにかかった時間、訂正回数、実行画面のコピー（もしくはその履歴）、（自分なりの）このプログラムの説明、5回(以上)ゲームした時の正解までの入力回数