

言い訳しない C++ 入門

開発駆動コース / 坂井ゼミ : 東 佑輝

どんな本？

- 対象者：

プログラミングをこれから始めたい人

低レイヤーに興味がある人

どんな本？

- 内容：

- 初学者に対してブラックボックスを作らない C++ 入門本
- コンピュータの仕組みを理解しながら C++ でのプログラミング手法を学んでいく

どんな本？

- 内容：
 - 初学者に対してブラックボックスを作らない C++ 入門本
 - コンピュータの仕組みを理解しながら C++ でのプログラミング手法を学んでいく
- 仕組みを知るのはたのしい！ これをモチベーションにしたプログラミング入門本がほしい

どんな本？

- 基本的には、低レイヤについて解説したい！というのが動機であるが、一応どういうときに役立つか考えてみる

どんな本？

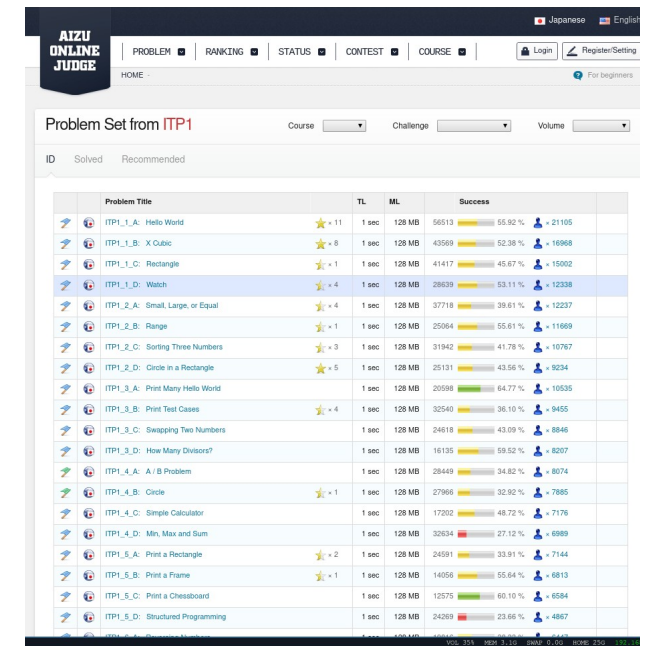
- 例えば ... (実際にあった例)
- 高専に入学し、初めて C++ に触る A 君



しーぷらぷらあああ

どんな本?

- とりあえず、C++ の基本文法はやってみた



The screenshot shows the AIZU ONLINE JUDGE website interface. At the top, there are navigation links for PROBLEM, RANKING, STATUS, CONTEST, and COURSE. Below this, there's a section for 'Problem Set from ITP1' with filters for Course, Challenge, and Volume. A table lists various problems with their titles, difficulty (TL), memory limit (ML), and success rates. The table includes columns for Problem Title, TL, ML, and Success. The problems listed are:

Problem Title	TL	ML	Success
ITP1_1_A: Hello World	★ × 11	1 sec	128 MB
ITP1_1_B: X O Game	★ × 6	1 sec	128 MB
ITP1_1_C: Rectangle	★ × 1	1 sec	128 MB
ITP1_1_D: Watch	★ × 4	1 sec	128 MB
ITP1_2_A: Small, Large, or Equal	★ × 4	1 sec	128 MB
ITP1_2_B: Range	★ × 1	1 sec	128 MB
ITP1_2_C: Sorting Three Numbers	★ × 3	1 sec	128 MB
ITP1_2_D: Circle in a Rectangle	★ × 5	1 sec	128 MB
ITP1_3_A: Print Many Hello World		1 sec	128 MB
ITP1_3_B: Print Test Cases	★ × 4	1 sec	128 MB
ITP1_3_C: Swapping Two Numbers		1 sec	128 MB
ITP1_3_D: How Many Divisors?		1 sec	128 MB
ITP1_4_A: A / B Problem		1 sec	128 MB
ITP1_4_B: Circle	★ × 1	1 sec	128 MB
ITP1_4_C: Simple Calculator		1 sec	128 MB
ITP1_4_D: Min, Max and Sum		1 sec	128 MB
ITP1_5_A: Print a Rectangle	★ × 2	1 sec	128 MB
ITP1_5_B: Print a Frame	★ × 1	1 sec	128 MB
ITP1_5_C: Print a Chessboard		1 sec	128 MB
ITP1_5_D: Structured Programming		1 sec	128 MB

- 夏休み前、先輩 K から仕事を割り当てられた
- とりあえず大切そうなコードを読んで見る

どんな本?

```
Kunugida::Kunugida(QWidget *parent) :  
    QMainWindow(parent),  
    ui(new Ui::Kunugida)  
{  
    ui->setupUi(this);  
    logger = spdlog::get("Kunugida");  
  
    QDir("../..procon2017-comp").mkdir("CSV");  
  
    // imageRecognitonTest();  
  
    connect(ui->RunButton, &QPushButton::clicked, this, &Kunugida::clickedRunButton);  
  
    connect(this, SIGNAL(requestCSV()), this, SLOT(getCSV()));  
    connect(this, SIGNAL(requestpostCSV()), this, SLOT(postCSV()));  
    manager = new QNetworkAccessManager(this);  
  
    board = std::make_shared<NeoAnswerBoard>();  
    board->show();  
    board->showMaximized();  
}
```


どんな本?

```
Kunugida::Kunugida(QWidget *parent) :  
    QMainWindow(parent),  
    ui(new Ui::Kunugida)  
{  
    ui->setupUi(this);  
    logger = spdlog::get("Kunugida");  
  
    QDir("../..procon2017-comp").mkdir("CSV");  
    // imageRecognitonTest();  
  
    connect(ui->RunButton, &QPushButton::clicked, this, &Kunugida::clickedRunButton);  
  
    connect(this, SIGNAL(requestCSV()), this, SLOT(getCSV()));  
    connect(this, SIGNAL(requestpostCSV()), this, SLOT(postCSV()));  
    manager = new QNetworkAccessManager(this);  
  
    board = std::make_shared<NeoAnswerBoard>();  
    board->show();  
    board->showMaximized();  
}
```

どんな本?

```
Kunugida::Kunugida(QWidget *parent) :  
    QMainWindow(parent),  
    ui(new Ui::Kunugida)           allow operator  
{  
    ui->setupUi(this);             this pointer  
    logger = spdlog::get("Kunugida");  
  
    QDir("../..../procon2017-comp").mkdir("CSV");  
    // imageRecognitonTest();      dot operator  
    connect(ui->RunButton, &QPushButton::clicked, this, &Kunugida::clickedRunButton);  
    connect(this, SIGNAL(requestCSV()), this, SLOT(getCSV()));  
    connect(this, SIGNAL(requestpostCSV()), this, SLOT(postCSV()));  
    manager = new QNetworkAccessManager(this); ← new operator  
                                                    (smart pointer)  
    board = std::make_shared<NeoAnswerBoard>(); ← make_shared()  
    board->show();                               function  
    board->showMaximized();                       (smart pointer)
```

どんな本？

ナニガ` ナンタ` カ <(`° д° `)> ワケワカラン !!

どんな本？

• 問題点

- よくある入門のやつで書くコードと実際に大規模開発で書くコードは全然違う
- スマートポインタのありがたみはメモリ管理をしらないとよくわからない
 - こういった不十分な理解のままプログラミングをすすめてしまうと、`unique_pointer` で十分なところを `shared_pointer` にしてしまい、メモリリークを起こすなどといったバグにもつながる（あるある）

どんな本？

- **本書による解決策**

- 実用ベースのコード例をもとに解説をする
- 低レイヤを知っていれば、メモリ管理やスマポのありがたさもすぐにわかる

目次・内容紹介

- ・ ポスターの原稿

苦勞している点

- 前提知識の線引き
- 構成
-

執筆環境について

- **Re:VIEW** を使ってます。
→ めっちゃ綺麗に出力できる

The image shows a screenshot of the Visual Studio Code editor with a C program open. The code is a simple program to calculate the sum of 3 and 4, but it includes extensive comments in Japanese explaining the process and the use of the printf() function. The code is as follows:

```
16  a b c d e f g h i j k l m n o p q r s t u v w x y z
17  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
18  0 1 2 3 4 5 6 7 8 9
19  _ ( ) [ ] # { } < > % ; : . ? * + - / ^ & | ~ ! , \ " '
20  //)
21
22  == 変数との出会い
23  ところで、<chap>(introduction)では文字列を出力しただけで、main()関数内で計算はしませんでした。
24  しかし、プログラミングといえば計算です。コンピュータの本質は計算機なので、計算をするべきなのです。
25
26  早速計算を試みましょう。
27
28  //list[calc][3+4を計算するためのソースコード]
29  int main()
30  {
31      int result;
32      result = 3 + 4;
33      printf("%d\n", result);
34      return 0;
35  }
36  //)
37
38  これをコンパイルして実行すると、3 + 4の答えが出力されるはずです。
39
40  ハローワールドのときは違ってprintf()関数の中には変な記号が入っています。
41  これは、フォーマット指定子といって、任意の数を文字として出力させたいときに使います。
42
43  == printf()関数のフォーマット
44  ここで、printf()関数のフォーマットについてまとめて置きます。
45
46  まず前提として、printf()関数では第1引数以フォーマットを指定し、第2引数以降にそのフォーマットで使用する任意の数を順番にとり第1引数で指定したフォーマットで文字を出力してくれます。フォーマットを指定するときには、「%」から始まる記号を使います。
47
48  //table[binary-decimal-hexadecimal][printf()で扱える代表的なフォーマット指定子]
49  指定子  型  説明  例
50  -----
51  %d, %i  int    int型の整数を10進数で出力する。      31415
52  %o      unsigned int unsigned int型の整数を8進数で出力する。      75267
53  %u      unsigned int unsigned int型の整数を10進数で出力する。      31415
54  %x, %X  unsigned int unsigned int型の整数を16進数で出力する。      .
55  .        なおxを大文字にした場合は、出力されるアルファベットが大文字になる。      7ab7
56  %e, %E  double    double型の実数を指数表記で出力する。      .
57  .        なおeを大文字にした場合は、仮数部と指数部の間のeが大文字になる。      3.141500e+00
58  %f, %F  double    double型の実数を出力する。      3.141500
59  %g, %G  double    double型の実数をいい感じに出力する。      3.1415
60  %c      unsigned char unsigned char型の整数を文字にして出力する。      A
61  %s      unsigned char* unsigned char*型を文字列にして出力する。      ABC
62  //)
63
```

The editor interface includes a sidebar on the left with icons for Explorer, Search, and Run and Debug. The status bar at the bottom shows system information like 'Ln 76, Col 1, Spaces: 4, UTF-8, LF, Re-VIEW' and a feedback survey prompt: 'Do you mind taking a quick feedback survey?'. The survey has 'Take Survey' and 'Remind Me later' buttons.

どんな本？

- ・ 小文字と大文字の違い

—

どんな本？

- ・ 小文字と大文字の違い

—

どんな本？

- ・ 小文字と大文字の違い

—

どんな本？

- ・ 小文字と大文字の違い

—

どんな本？

- ・ 小文字と大文字の違い

—

終

