

PL04 プログラミングC++2021 授業概要

成瀬継太郎

naruse@u-aizu.ac.jp

授業の時間と形式

- 講義: 毎週月・木曜日の13:20-15:00(大講義室), 担当: 成瀬(CS, IT-SPR)
- 演習: 同じ日の15:10-16:00(演習室5, 6), 担当: 成瀬(CS), 中村啓太(IT-SPR)
- 本年度(2021年度)は状況に応じて対面授業と遠隔授業の使い分けを行う
 - 10月4日(月)から10月15日(金)までは遠隔講義を行う
 - それ以降は基本対面授業だが, 状況に応じて変更する可能性がある
- 遠隔授業のzoom会議室情報
 - Topic: Programming C++
 - Time: 13:20-16:00 every week on Mon, Thu, from Oct 4, 2021 until Nov 25, 2021
 - <https://us06web.zoom.us/j/84786672218?pwd=cE01RkhxcWZnOFNGc kJJSE1XZUx0QT09>
 - Meeting ID: 847 8667 2218
 - Passcode: 941325

授業サイト

- 授業のWebページは大学共通のmoodleを使います
 - 講義: 21講義 PL04 プログラミングC++
 - <https://elms.u-aizu.ac.jp/course/view.php?id=4780>
 - 講義のハンドアウト
 - クイズの採点
 - 期末試験情報など
 - 演習: 21演習 PL04 プログラミングC++ [演]
 - <https://elms.u-aizu.ac.jp/course/view.php?id=4781>
 - 演習課題
 - 演習の採点
- 授業のメーリングリスト
 - 講義: 18-1405-PL04-LTh@course.u-aizu.ac.jp
 - 演習(成瀬クラス) : 18-1407-PL04-std5@course.u-aizu.ac.jp
 - 演習(中村クラス) : 18-1407-PL04-std6@course.u-aizu.ac.jp

評価

- 演習: 50%
 - 必修13
- 期末試験: 20%
- 出席とクイズ: 30%
 - クイズの提出回数が全体の2/3に満たない場合は無条件に不合格とする
→10回の提出が必要
 - クイズは授業開始中に回答する
 - 単位を取得するには, 演習, 期末試験, 出席とクイズのすべてを合格する必要がある

演習課題の提出

- 提出方法: 授業サイトからアップロード
- 提出期限: 1週間まで
 - 授業が月曜として翌週火曜の日付のものは遅れたとみなし受け付けない
- 注意事項
 - 期限までに提出すること→遅れると採点しない
 - ただし, 提出後の再提出は認める
(学習効果を高めるためにも, 採点後に積極的に再提出してください)

演習採点基準

- 各演習は10点満点
- 採点基準: 減点法(最低0点)
 - コンパイルが通らない: -10
 - コンパイルは通るが結果が間違い: -4から-8
 - コメントがない, 不適切: -2

(アンケート)演習の計算機環境について

- 皆さんが普段使っている計算機で演習を課題を行うことが多いと思いますので、質問です。
皆さんのOSは何ですか？C++のコンパイラは何を使っていますか？
- Windows
 - Visual Studio 2019
 - Ubuntu / WSL2(Windows Subsystem for Linux)
 - Ubuntu / VMWare Player
- MacOS
- Ubuntu
- CentOS

授業予定前半 (本年度は月曜日の休講が多いです)

講義回	講義日	内容
1	10月4日 (月)	授業概要, 第0章「さあ始めよう」第1章「stringを使う」
2	10月7日 (木)	第2章「ループとカウンタ」
休講	10月11日(月)	大学祭翌日のため
3	10月14日 (木)	第3章「たくさんのデータを扱う」
休講	10月18日(月)	ワクチン接種翌日のため
4	10月21日 (木)	第4章「プログラムとデータの構成」
休講	10月25日(月)	ワクチン接種翌日のため
5	10月28日 (木)	第5章「シーケンシャルコンテナとstringの解析」
6	11月1日 (月)	第6章「ライブラリのアルゴリズムを使う」
7	11月4日 (木)	第7章「連想コンテナを使う」
8	11月8日 (月)	第8章「ジェネリック関数を書く」

授業予定後半

講義回	講義日	内容
9	11月11日 (木)	第9章「新しい型を定義する」
10	11月15日 (月)	第10章「メモリ管理と低レベルのデータ構造」
11	11月18日 (木)	第11章「抽象データ型を定義する」
12	11月22日 (月)	第12章「値のように振舞うクラス」
13	11月25日 (木)	第13章「継承と動的結合を使う」
14	11月29日 (月)	第14章「メモリを(ほとんど)自動的に管理する」
期末試験	未定	期末試験期間中に実施する

この授業の到達点

- C++のコードを読める・書けるようになること
→画像処理(OpenCV), 点群データ処理(Point Cloud Library), 行列演算(Eigen)などを利用できるようになる
- プログラムの開発＝データ構造の設計＋アルゴリズムの適用ができるようになること
- プログラムのテストができるようになること
→思い付きコード修正から系統的なテストへ

C++とは

- C言語と互換性あり
 - C言語のプログラムはC++でコンパイル可能
- C言語を「オブジェクト指向」化したもの
 - 自分で作った「クラス」が組み込みの変数や関数と同じように使えるようになる
 - 「=」や「+」が自分で定義できる
 - 使いやすい道具を作りやすい
- グループによる開発に向いている

C++を学ぶとは

- C++の組み込みのデータ形式を学ぶ
- 標準ライブラリ関数(STL)を学ぶ
- クラスの概念を用いたプログラム設計ができること
 - 何をクラス変数にするか
 - 何をメンバ関数にするか
 - どう設計すれば使いやすいクラスになるか
 - クラスのテストの仕方を学ぶ

本講義で学ぶC++のバージョン

通称	規格出版日	概要
C++98	1998-09-01	最初の標準化
C++11	2011-09-01	現在のC++の基本形. 本講義ではこの規格のC++を学ぶ
C++14	2014-12-15	C++11のマイナーアップデートで, いくつかの機能が追加された
C++17	2017-12	新しい機能が追加され, 一部の古い機能が廃止された 便利な機能が豊富で, 最近はこれが標準になりつつある. しかし, 本講義ではC++の原理を学ぶために, あえて古いC++11を学ぶ

C++の二つの面

- C++ as Better C
 - C++言語はライブラリが追加された使いやすいC言語としてつかうことができる
 - string, Standard Template Library (STL)
 - ポインタ, リファレンス(参照), イテレータ
 - 講義の第1回から第8回まで
- C++ as Objective-oriented C
 - 一方、C++言語はC言語をオブジェクト指向化したものとも見える
 - Class
 - 講義の第9回から第14回まで

第0章「さあ始めよう」

主なトピック

- コンパイルの方法
- コメントの書き方

最初のC++のソース

```
/*  
    hello.cc  
    概要：最初のC++プログラム  
    Author: Keitaro Naruse  
    Date: 2021-10-01  
*/  
  
#include <iostream>  
  
int main()  
{  
    // 画面に文字列を出力し，改行  
    std::cout << "Hello, World!" << std::endl;  
  
    return 0;  
}
```


ソースの編集とコンパイルの方法

- C言語とほぼ同じ
- エディタで編集
 - C++のソースファイルの拡張子は.cc .cppなど
- コマンドラインからコンパイル
 - `> g++ hello.cc`
 - 成功するとa.outという実行形式が生成
 - `> ./a.out` で実行
 - `g++ -o hello hello.cc` とすると, helloという実行形式ができる

ソースを見ると(1)

- 1から6行目と12行目: コメント行
 - // で始まる行は行末までコメント
 - C言語で使われている /* */ も使用可能
- 8行目: `#include <iostream>`
 - プログラムの中で標準ライブラリを利用するときに使用
 - この場合は*iostream*に含まれる標準ライブラリを使うとき

ソースを見ると(2)

- 10行目: `int main()`
 - メイン関数, 必ず実行される
 - `int`型(整数型)の値を環境(OS)に返す
 - この場合は, 外部から受け取るデータはない
- 11,16行目: `{ }`
 - 中カッコで囲まれたのが, ブロック
 - ブロックの中に書かれるのが, ステートメント(文)

ソースを見ると(3)

- 13行目: `std::cout << "Hello, world!" << std::endl;`
 - 標準出力(画面)にHello,world!という文字列を出力する
 - `std::cout` 標準出力ストリーム, 画面
 - `std::endl` 改行文字
 - `std::` (標準の)名前空間
 - `<<` 出力演算子
 - 文字列は`""`(ダブルクォーテーション)で囲う
 - 文末は`;`(セミicolon)

ソースを見ると(4)

- 15行目: `return 0;`
 - 関数の戻り値として0を返す(この場合は環境に)
- 自由形式
 - 文字列と文字列の間の改行や空白は自由に置いて良い
 - ソースが見やすくなるようにインデント

この科目の「美しい」ソースコード

```
/*  
    hello.cc  
    概要：最初のC++プログラム  
    Author: Keitaro Naruse  
    Date: 2020-10-01  
*/  
  
#include <iostream>  
  
int main()  
{  
    // 画面に文字列を出力し、改行  
    std::cout << "Hello,  
World!" << std::endl;  
  
    return 0;  
}
```

- 演習はこのスタイルに従ってください
 - ソースの先頭にプログラムの概要, 作者, 作成日を表すコメント
 - 見やすくなるように改行を多用
 - インデントの字下げは4文字
 - 処理の概要がわかるようなコメント