

ものづくり基礎工学

ようこそ情報工学分野へ

J 1. プログラミングの基本

準備

- ◆ 情報のHPからリンクをたどる



ものづくりのページへ

スライドを開く

※スクリーンが見えない人はPCで確認しながら進める

- ◆ 情報ものづくりのHPとスライドを2つ開いておく

はじめに

- ◆ 担当
- ◆ 情報工学とものづくり

詳細はガイダンスで聞いていると思います



情報工学科の研究室は主に3つの柱

知識：文字認識，人工知能など

通信：携帯，ネットワーク，セキュリティなど

制御：ロボット，音響，環境など

現在は，どこでも「情報」に関係する技術が使われている

※希望学科でなくとも，情報の内容はどこかで触れることがある。

将来は全く異なる分野でも，知っておくとよい。

コメント：ものづくりはほんの一端。

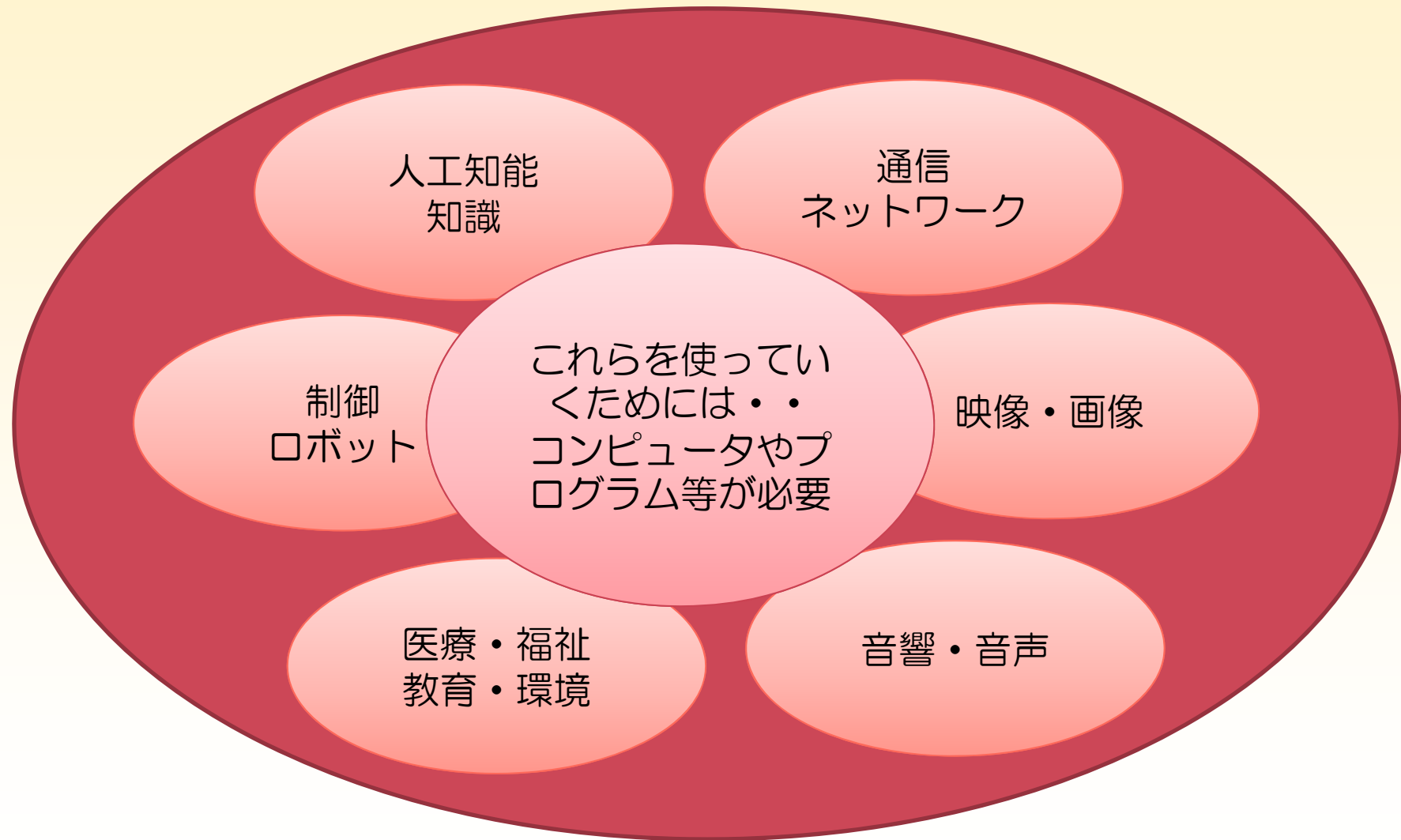
学科を選ぶときは，**どんな科目があるか**，**どんな内容なのか**，**どんな卒研があるか**，**進路があるか**などしっかり調査すること

情報工学科での
パソコン，プログラムは
ものづくりのための
1つのツール（手段）
ではない

重要なので勉強します。
でもパソコンやプログラムが
メインではありません。

冷蔵庫，炊飯器
家電ネットワーク
携帯電話・スマホ
自動車などなど

情報工学



情報工学に関連する分野はとても広い！

今後の予定

- ◆ プログラミングの基礎（CG） 最初2回
でも実際はプログラミングの部分は少し！

プログラムが大変というよりも
数学(座標)の部分で苦労する！？

どの学科でも・・・

数学や物理が重要になることを知ってほしい

今後の予定

- ◆ マイクロコンピュータで制御 2回

実際のコンピュータをプログラミングして使ってみる（今回は基礎。実際には、利用者のアイデアしだいでいろいろな使い方が可能）。

- ◆ 音の分析 最後1回

いつも何気なく聞いている音について、少し工学的な立場から見てみよう。

実験を進める上での注意

- ◆ 「進捗報告書」と「今日のまとめ」

毎回の実験終了後に提出（内容は後で指示）

- ◆ プログラムは実験終了時にチェック

きちんと指定通りに作成すること

- ◆ 実験の日（金曜日）は予定を入れないように

- ◆ 情報分野の合格には

- 1) すべての回に出席

- 2) すべての進捗報告が提出&内容も十分

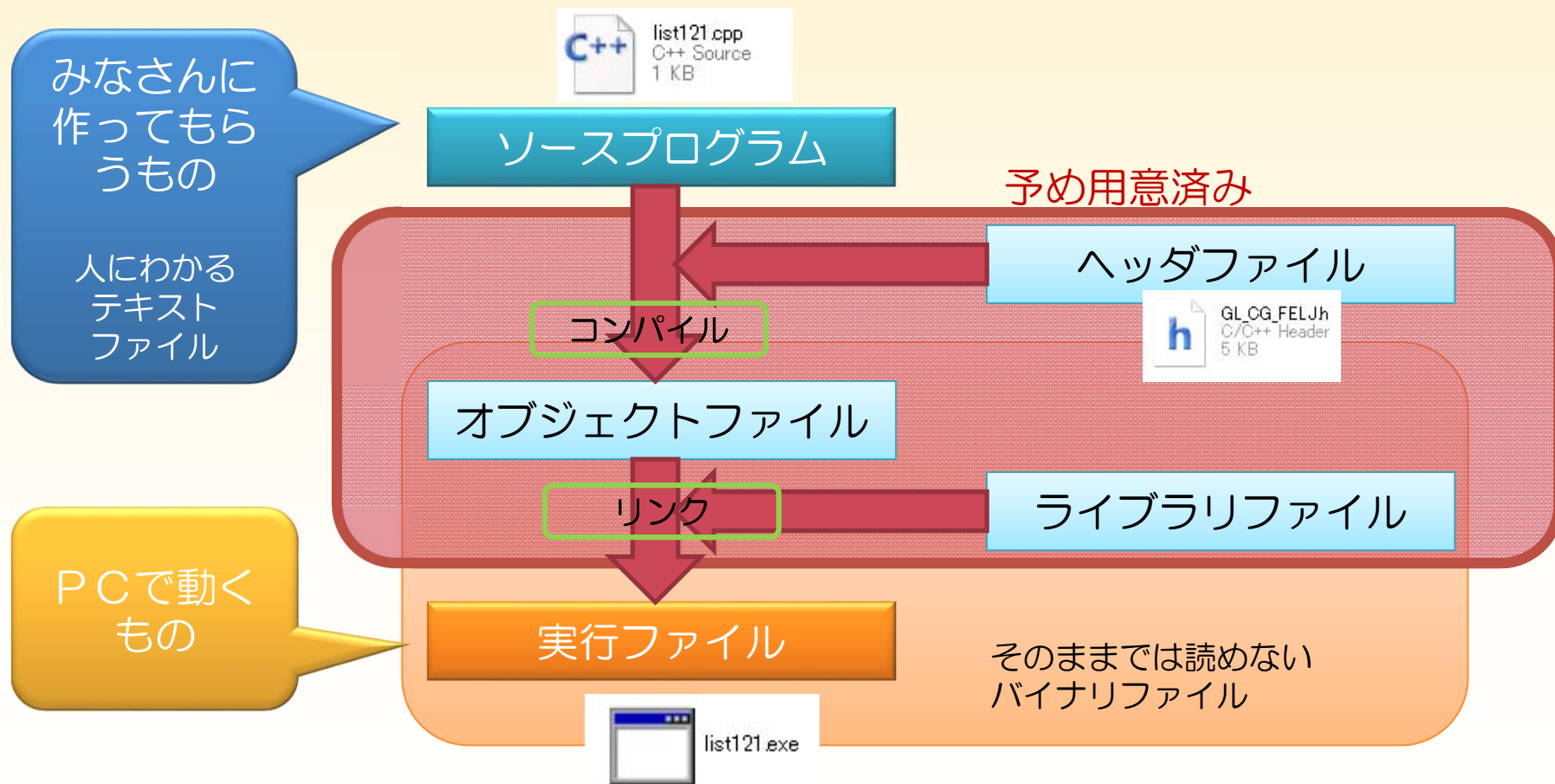
- 3) 指定されたプログラムがすべて作成済み

- ◆ 熱などで休んだ

休む必要がある場合など、すぐに担当者に連絡する。

J1.プログラミングの基本

◆ 実験でのプログラム作成手順



いっしょにやってみてください

ファイルの準備

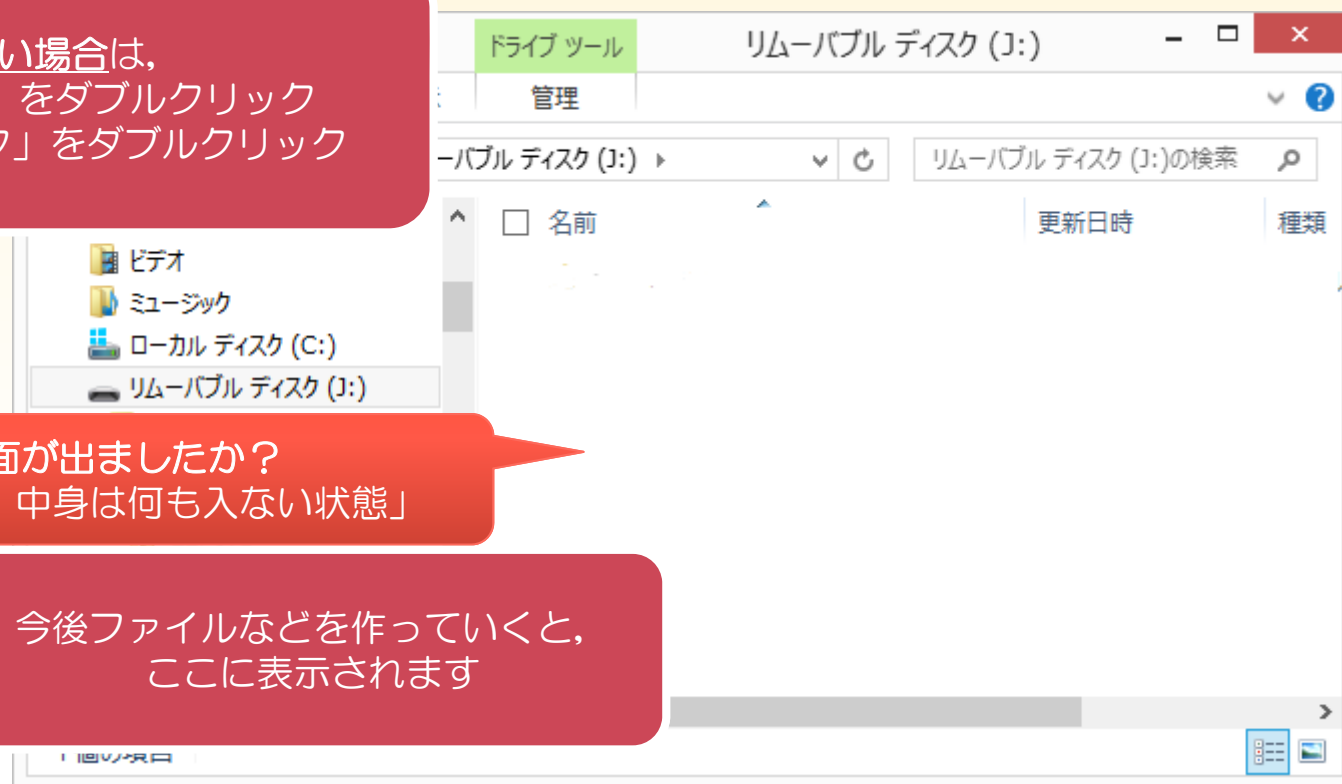
◆ USBメモリをパソコンへ

①このような画面が出てきたら
「フォルダを開いてファイルを表示する」を
選択しOK

画面が出てこない場合は、
デスクトップ左の「PC」をダブルクリック
「J:リムーバブルディスク」をダブルクリック

②このような画面が出ましたか？
「新品のUSBメモリなら、中身は何も入らない状態」

今後ファイルなどを作っていくと、
ここに表示されます



いっしょにやってみてください

ファイルの準備

◆ ファイルのダウンロード→USBメモリへ

プログラムでいろいろな模様を描こう

共通に必要なとされるファイル

本実験を行なうためには、各種の関数について記述したヘッダファイル、ソースファイル及びコンパイルオプションファイルである。次のリンクよりダウンロードし、展開しておく必要がある。

- 「プログラムでいろいろな模様を描こう」で必要とされるファイルをまとめたもの (LZH)

演習室のPCにはこれらのファイルがインストールされていない

J1. プログラミングの基礎

- 説明スライド (HTML)
- 説明スライド (PDF)

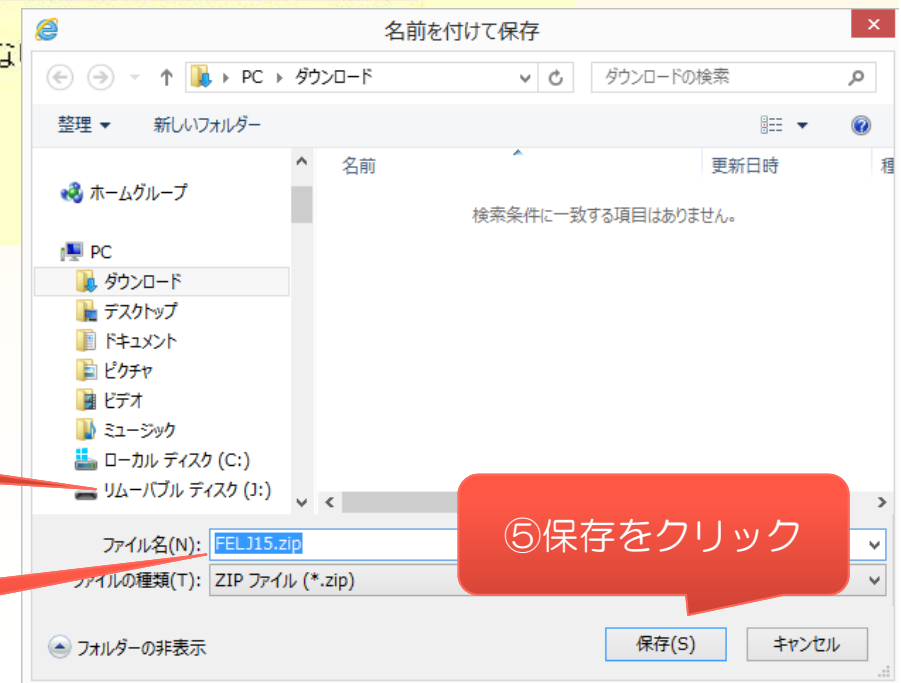
①左クリック

②「対象をファイルに保存」
をクリック

③「リムーバブルディスク (J:)」を選択

④FELJxx.zipを確認

⑤保存をクリック



いっしょにやってみてください

ファイルの準備

- ◆ ファイルの解凍
「リムーバブルディスク（J）」の中に
FELJxx.zip
があるので、ダブルクリックすると解凍される。
（FELJxxというフォルダができており、その中に
さらにFELJxxという同じ名前のフォルダがある）

この中には3つのファイルがある

- GL_CG_FELJ.cpp
- GL_CG_FELJ.h

自分で作るもの以外に必要なファイル（直接は使いません）

- vccnet.cmd

プログラムを動かす時にみなさんが利用するファイル

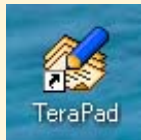
- list121.cpp
- list231.cpp

サンプルソースファイル（みなさんが作っていくもの）

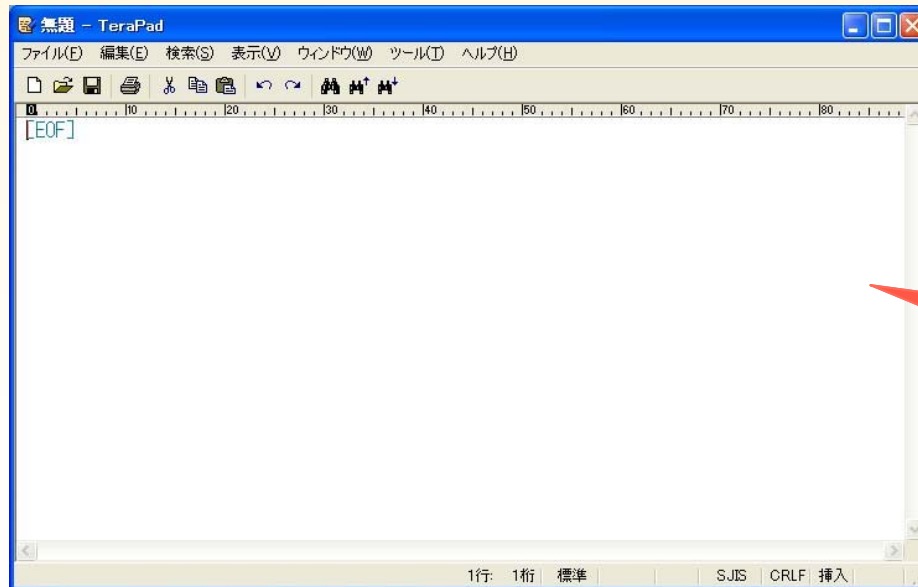
いっしょにやってみてください

プログラムの入力

◆ テキストエディタの起動



Terapad(テラパッド)を起動する



ここにプログラム
を記入していく

プログラムを入力してみましょう

プログラム入力時の注意

◆ リスト1.2.1を入力（サンプルあり）

#はShiftキーを押しながら
3キーを押す

全角・半角，大文字・小文字
に気をつけて入力

```
/* 1つの長方形の描画 */  
#include <stdio.h>  
#include <math.h>  
#include <GL/glut.h>  
#include "GL_CG_FELJ.h"  
  
void userdraw(void)  
{  
    fillRectangle(-180, 100, 200, -100, RGB(0, 0.5, 0.9));  
}
```

このスペースは「Tab」キーで入力

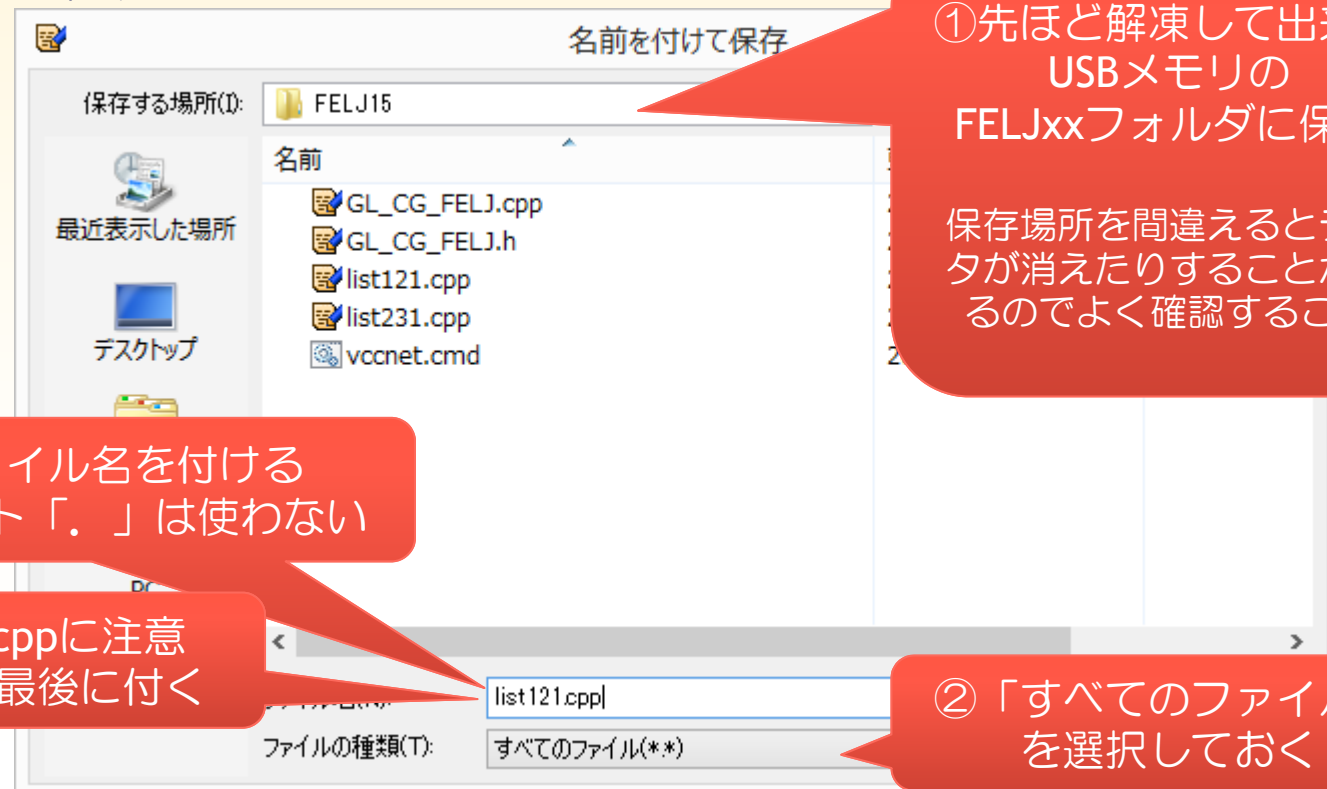
最後にセミコロンを入力

いっしょにやってみてください

プログラムの保存

◆ リスト1.2.1を入力後

Terapadの左上「ファイル」から「名前を付けて保存」を選択



①先ほど解凍して出来た
USBメモリの
FELJxxフォルダに保存

保存場所を間違えるとデー
タが消えたりすることがあ
るのでよく確認すること

③任意のファイル名を付ける
スペース, ドット「.」は使わない

④ファイル名.cppに注意
「.cpp」が必ず最後に付く

②「すべてのファイル」
を選択しておく

いっしょにやってみてください

プログラムのコンパイルと実行

◆ FELJフォルダの中の



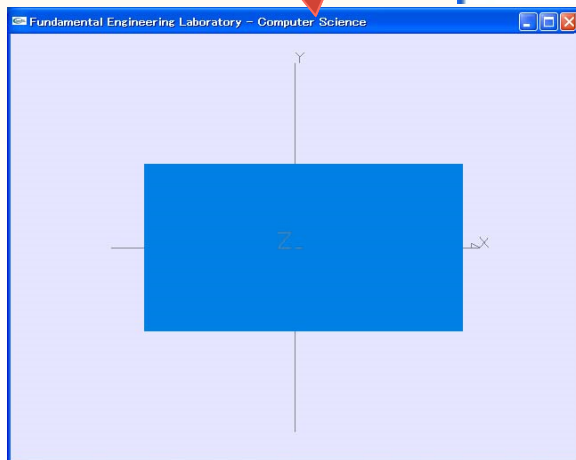
③実行ファイルをダブルクリックするとCGが表示される（但し、左のデスクトップ画面で行います）

①作成したファイルをドラッグ&ドロップをする

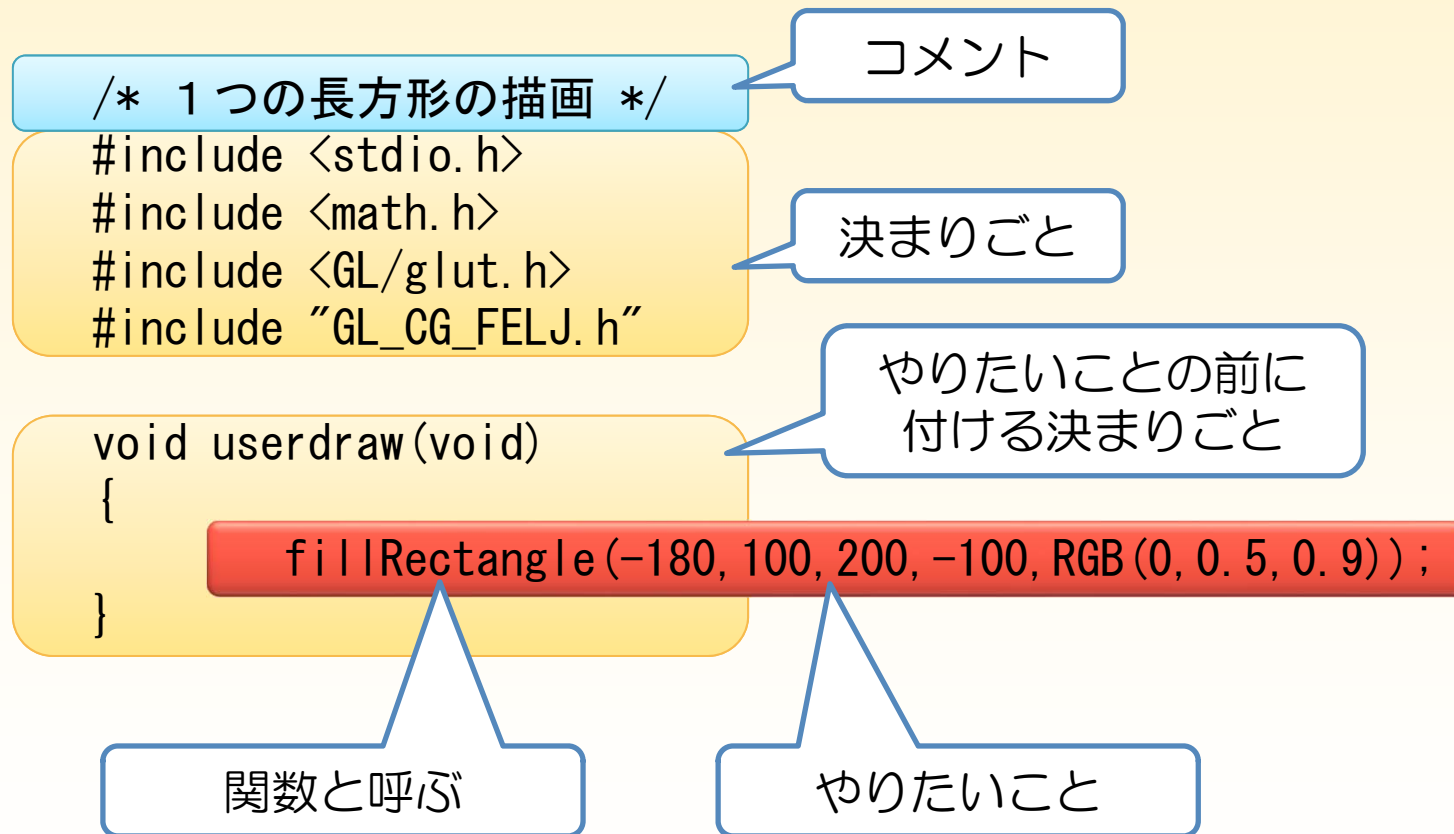
②エラーがなければ実行ファイルが出来る

④CGを確認したら×で消すようにしてください。

せっかくプログラムの修正をしても、同じファイル名のCGが表示されていると、修正結果が反映されません



各部の役割

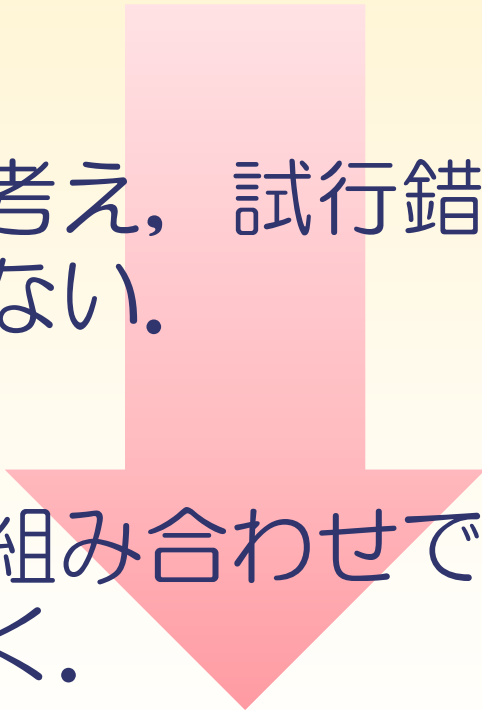


ただしここでの決まりごとは、実際のC言語とは若干異なるので注意

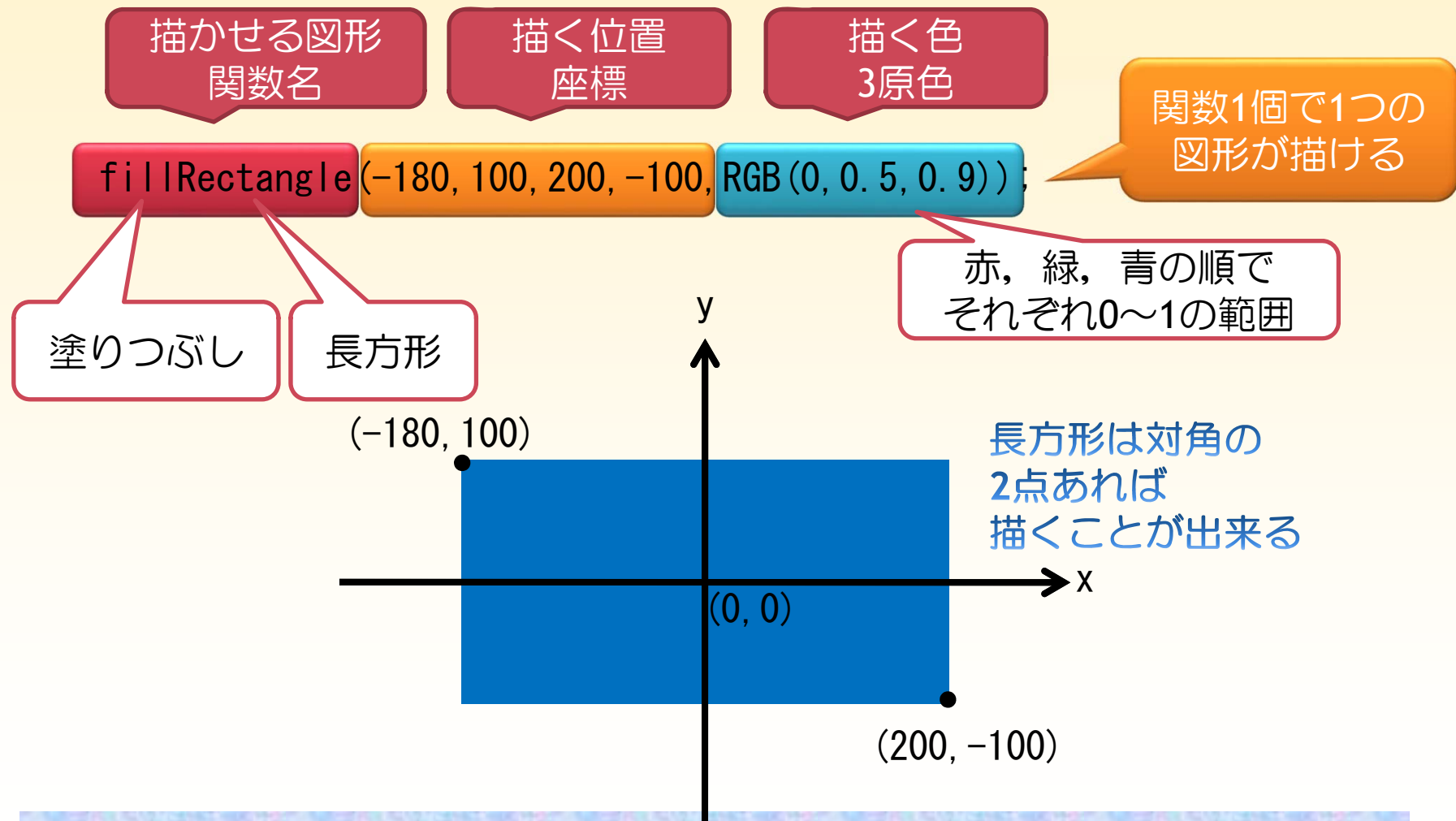
ここまでの注意

- ◆ プログラムの保存場所は間違えないように
（なれていない人はUSBメモリ「リムーバブルディスク」に保存
- ◆ PCの「デスクトップ」などでも可能. 最後に「FELJフォルダ」ごとUSBにコピーすればよい.
- ◆ データの管理は自己責任で（バックアップをこまめにとる）.

図形を書くための指示

- ◆ プログラムは論理的な組み立てが重要
 - ◆ 座標をよく考え、試行錯誤的なプログラミングはしない.
 - ◆ 基本図形の組み合わせで、複雑な図形を構成していく.
- 

長方形を描く



色の指定はWEBの「[光の3原色による色の合成](#)」をクリックして参考にする
(いくつか数値が並んでいるが, 0から1の範囲の数値の並びを利用する)

ここで演習課題

- ◆ 課題1.1の(1), (2)をやってみてください。
文章をよく読み, きちんと座標を出してください.
- ◆ 課題1.2はスキップします.
- ◆ 時間に余裕がある人は, 三角形, 円, 楕円の描画を行ってみてください.
- ◆ 座標の計算で小数点がでてくる場合は
小数点以下を四捨五入か, 切り捨て
したのでよいです (微少な数値の違いは, 画面上ではほとんど分からない).

初めに座標を紙で計算してから, プログラミング!

※windows「スタート」画面の「電卓」を利用しても使えます

ヒントは次のページ

ここでの課題のヒント

- ◆ 一行（1つの関数）で1つの長方形（図形）が描けた。3つ描くにはどうすれば？
- ◆ 描く順番が大切。小さい図形の上に大きい図形を重ねると隠れてしまう。
- ◆ 課題1.1(2)は大きさ（比を考えて！）

例と「位置・大きさ」が違うと不可

注意：今後の課題で、自分が作ったものがあるかの参考には
WEBの「スクリーンショット集」を参考
にしてください。

色の指定については、スクリーンショットを見ながら似た色を自分で探し、
「おおよそ同じ色」でかまいません

お昼の休憩時間に向けての注意

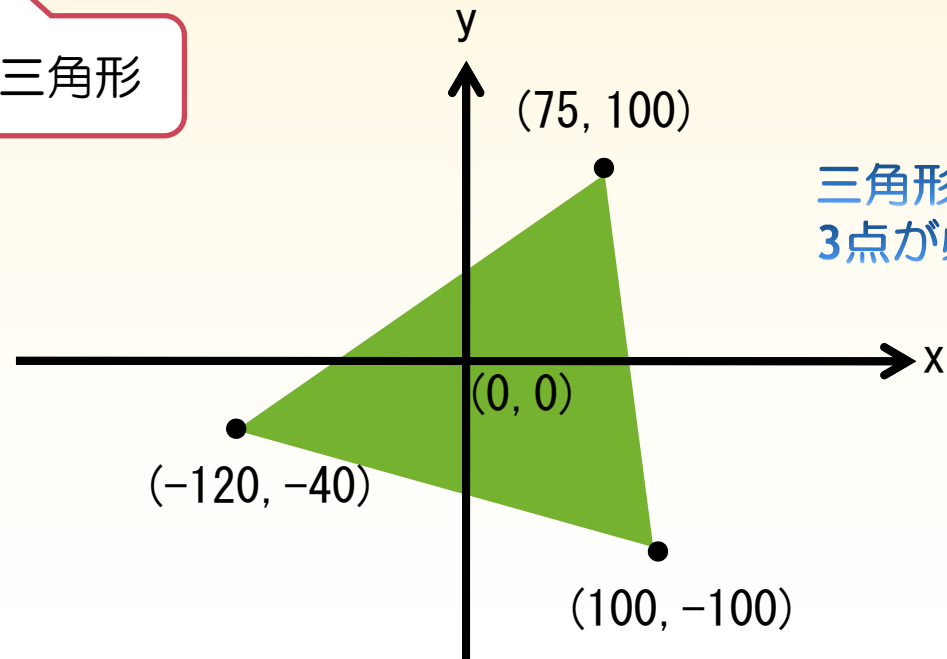
- ◆ PCは電源を切ると、作成したデータも消えますので、必ずUSBメモリに保存しておくこと.
- ◆ 自分の荷物・貴重品は置いたままにしないこと.
- ◆ 12:50までに戻る. 各午後の自準備をしておく.

三角形を描く

```
fillTriangle(75, 100, 100, -100, -120, -40, RGB(0, 0.8, 0));
```

塗りつぶし

三角形



三角形は頂点の
3点が必要

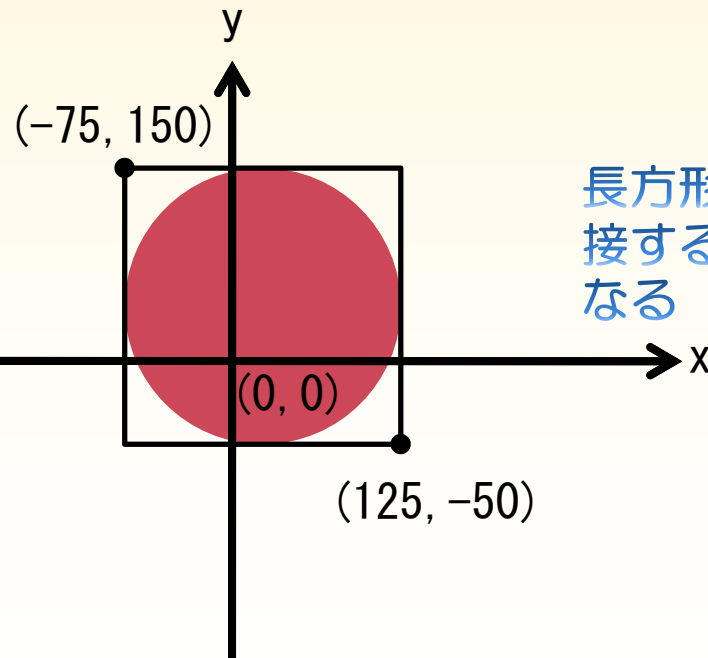
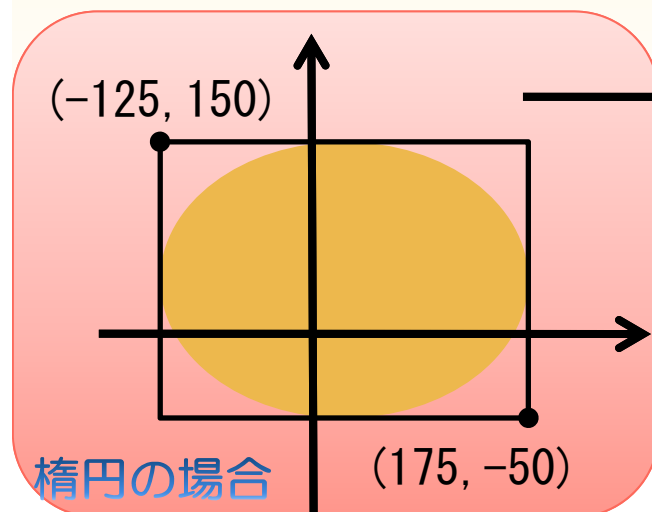
円・楕円を描く

```
fillEllipse(-75, 150, 125, -50, RGB(1, 0, 0));
```

塗りつぶし

楕円

長方形、正方形に内接する円を描く形になる



線画を描く

関数名のfillをdrawに変更することで、
中を塗りつぶさずに線だけを描くことができる

fillRectangle	→	drawRectangle
fillTriangle	→	drawTriangle
fillEllipse	→	drawEllipse

塗りつぶし

線画

直線を描く関数も用意されている

drawLine

練習と報告書

- ◆ 練習でリスト1.4.1～1.4.4を行ってみましょう.
- ◆ 課題1.3が終わったら進捗報告書を取りに来てください.
- ◆ 課題1.5を行ってください
(レポートの指示をよく読むこと. スクリーンショットとは 異なることがあります).
(課題1.4はスキップ).
- ◆ レポートが出来たらチェックをしてもらってください (CGのチェックも行います).

※ここで終了です. 時間があれば課題1.6もやってみましょう.

最後のヒント

- ◆ 課題1.3の六角形の図は
直角三角形（ $1:2:\sqrt{3}$ ）を見つけ出すとわかりやすい
座標の象限を考える
(軸に対象の座標は符号（±）を考えると簡単なことも)
- ◆ 課題1.5は「指示された1メモリのスケール，原点」
に注意.
 - メモリは1/10まで読むこと
 - 原点やスケールはレポートに書いています

これらは，レポートの座標と作成したCGが対応し
ていないと不可です