# 情報処理及び実習
# 第4回（4/24）

月曜日4,5-1時限（第1実習室）
火曜日2時限（第4実習室）4/17,4/24
鈴木良弥，丹沢勉
TA: 小佐野，佐藤

授業資料 http://www.ccn.yamanashi.ac.jp/~ysuzuki/JMinfo/index.html

## 前回の実習
## C言語プログラミング入門の準備1

- Visual Studioの起動
- コンソールアプリの作り方
- C言語でのプログラミング
  - プログラミング入門ではないので，サンプルプログラムが理解できなくても良い

## 前回と今回の実習の目的

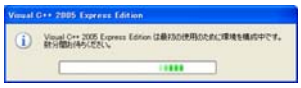
- Visual Studioの起動方法を覚える
- プロジェクトの作り方を覚える
- プログラムの実行の仕方を覚える
- C言語のプログラムに慣れる
- ソースコードの字下げ（インデント）の必要性を理解し，活用する

## Visual Studioの使い方
## （プログラミング入門で使用）

- Visual Studio
  - Microsoftのプログラミング用統合環境
  - C++(C), C# , Basicなどの複数言語で利用可能
  - プログラミングに必要な機能が満載
    - レイアウト作成
    - コード補間機能（IntelliSense）
    - コンパイラ
    - リンカ
    - デバッガ

## Visual Studioの起動

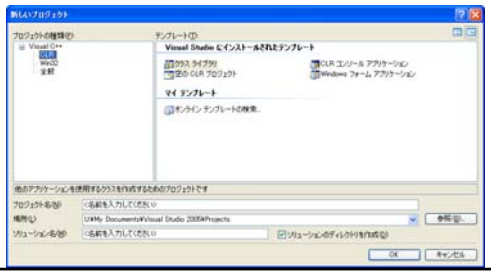
- 授業で使用するVisual Studio：
  - Visual Studio 2005 Express
- スタート→すべてのプログラム→Visual C++ 2005 Express Edition→Microsoft Visual C++ 2005 Express Edition




## Visual Studioの起動

## プロジェクトの作成

- ファイル→新規作成→プロジェクト
- プロジェクトの種類でWin32を選び，テンプレートとして「Win32コンソールアプリケーションを選択し，プロジェクト名を入力



## ソースコード入力まで

- 追加のオプション：空のオプション
- 「完了」ボタンを押す
- ソリューションエクスプローラのソースファイルを右クリック→追加→新しい項目
- 新しい項目：C++ファイル(.cpp)を選択
- 下のファイル名入力欄に「～.c」と入力
- ソースコードを入力

## ビルド，実行

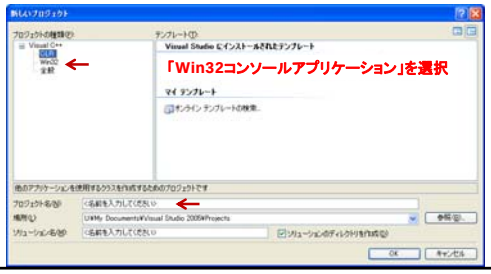
- メニューからビルド→ソリューションのビルド
- エラーが無いか確認
- デバッグ→デバッグの開始
- 画面がすぐに消えてしまう場合はブレイクポイントを追加
- デバッグ→デバッグの停止
- デバッグ→デバッグなしで開始　でもOK

## 実習1：プロジェクト　hello の実行

- hello, world を表示させる

## プロジェクト　hello の作成 1/5

- ファイル→新規作成→プロジェクト
- プロジェクトの種類でWin32を選び，テンプレートとして「Win32コンソールアプリケーションを選択し，プロジェクト名を入力

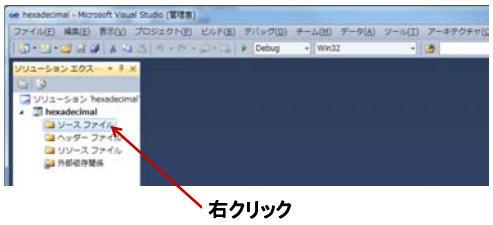



## プロジェクト hello の作成　2/5

- 「次へ」のボタンを押す
- 追加のオプション：空のオプションを選ぶ
- 「完了」ボタンを押す

## プロジェクト hello の作成　3/5

- ソリューションエクスプローラのソースファイルを右クリック→追加→新しい項目



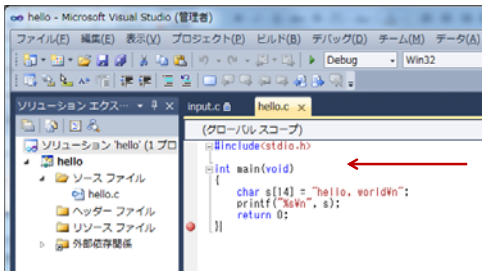

## プロジェクト hello の作成　4/5

- 新しい項目：C++ファイル(.cpp)を選択

- 下のファイル名入力欄に「hello.c」と入力

## プロジェクト hello の作成　5/5

- ソースコードを打ち込む



## ソースコード　hello.c
## プログラム1

```
#include<stdio.h>

int main(void)
{
    printf(“hello, world¥n");
    return 0;
}
```

ブレークポイント（一旦停止）　グレー部分を左クリック

## ソースコード　hello.c
## プログラム1

```
#include<stdio.h>   入出力用ヘッダファイルの組込み

int main(void)   main関数
{
        フォーマット付き出力
    printf(“hello, world¥n");
    return 0;
        返り値
}
```

ブレークポイント（一旦停止）　グレー部分を左クリック

## 実行結果

- 「hello, world」が一行目に表示されたウインドウが表示される
- 終了するには　デバッグ→デバッグの停止

## 作成したプロジェクトの格納場所

- プロジェクト（自分で変更しなければ）
  - ドキュメント ▸ Visual Studio 2005 ▸ Projects ▸ hello
- ソースファイル（hello.c)
  - hello ▸ hello ▸ hello.c

## プログラム　1e-1

```
#include<stdio.h>

int main(void)
{
    printf("Hello World¥n");
    return 0;
}
```

ここのセミコロン「;」を消してビルドするとどうなるか確認

## プログラム　1e-2

```
#include<stdio.h>

int main(void)
{
    printf("Hello World¥n");
    return 0;
}
```

ここの「}」を消してビルドするとどうなるか確認

## プログラム　1e-3

```
#include<stdio.h>

int main(void)
{
    printf("Hello World¥n");
    return 0;
//}
```

注釈マーク：このマーク以降はビルドに無視される

行頭に//をつけてビルドするとどうなるか確認

## ビルド時に下のウインドウに表示されるメッセージは重要

- 1:ビルドに成功しました。
- 1e-1:構文エラー：';' が 'return' の前にありません。
- 1e-2:左側 中かっこ '{' に対応するものが '...' で見つかる前に EOF が検出されました。
- 1e-3:左側 中かっこ '{' に対応するものが '...' で見つかる前に EOF が検出されました。

## プログラム　1m-1

```
#include<stdio.h>

int main(void)
{
    char s[14] = "hello, world";
    printf("%s¥n", s);
    return 0;
}
```

枠内を書き換える

## 変数(文字配列)s

```
char s[14] = "hello, world";
```

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| h | e | l | l | o | , | | w | o | r | l | d | ¥0 | |

ヌル文字
文字列の終端

## プログラム 1m-2

```
#include<stdio.h>

int main(void)
{
  char s[14] = "hello, world!";
  printf("%s¥n", s);
  return 0;
}
```

「！」を1つ追加して
ビルド→デバッグ開始

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| h | e | l | l | o | , | | w | o | r | l | d | ! | ¥0 |

## プログラム 1m-3

```
#include<stdio.h>

int main(void)
{
  char s[14] = "hello, world!!";
  printf("%s¥n", s);
  return 0;
}
```

「!!」を追加して
ビルド→デバッグ開始

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| h | e | l | l | o | , | | w | o | r | l | d | ! | ! | ¥0 |

## 配列のオーバーランに注意

- コンパイラは配列の範囲チェックをしてくれない
- 配列のオーバーランによる影響は予測不可能
- プログラマが責任をもってチェック

## ソリューション(プロジェクト)2
## 四則演算 arithmetic

- 新たにプロジェクトを作る

前回はここまで

## arithmetic.c (少し問題のあるプログラム)

```
#include<stdio.h>
int main(void)
{
  int a, b;
  printf("a="); scanf("%d", &a);
  printf("b="); scanf("%d", &b);
  printf("%d+%d=%d¥n", a, b, a+b);
  printf("%d-%d=%d¥n", a, b, a-b);
  printf("%d*%d=%d¥n", a, b, a*b);
  printf("%d/%d=%f¥n", a, b, (float)a/b);
  return 0;
}
```

キーボードにより整数を入力

%d: 整数を表示
%f: 実数を浮動小数点で表示

整数aを実数に変換

## 実習2

- プログラムを実行する.
  - a=5, b=3を入力
  - a=3, b=10
  - a=a, b=b
  - a=99, b=100
- プログラムを変更して実行
  - printf(“%d/%d=%f¥n”, a, b, (float)a/b); → printf(“%d/%d=%f¥n”, a, b, a/b);
  - a=99, b=100

## 実習3

- 10進数，16進数，2進数表記を並べて表示
- プロジェクト名：hexadecimal
- ソースファイル名：hexadecimal.c
- 次のページのプログラムを打ち込んだ後，いろいろ変更してどんなエラーや結果が表示されるかを観察

## hexadecimal.c

```
#include<stdio.h>
#include<stdlib.h>
#define BUFF_SIZE 10

int main(void)
{
  char buf[BUFF_SIZE];
  int i;
  printf("10進数   16進数       2進数¥n");
  for(i=0; i<=32; i++)
  {
    printf("    %02d       %02x    %08s¥n", i, i, itoa(i,buf,2));
  }
  return 0;
}
```

数値を2進数（文字列）に変換

## 記数法（十進数，十六進数，二進数）

- N進数：N種類の記号を使った数の表記法
  - 10進数：0～9の10種類の記号
  - 16進数：0～9,A～Fの16種類の記号
  - 2進数：0,1の2種類の記号
  - k+1桁目の記号はk桁目の同一記号のN倍を表す
    - 10進数の20は2の10倍
    - 16進数の20は2の16倍
    - 2進数の100000は10の$2*2*2*2*2=2^4$=16倍

## 2進数→10進数，16進数 hexadecimalを修正して確認

- 2進数の00101011
- $2^5+2^3+2^1+2^0=32+8+2+1$
- =43（10進数），2B（16進数）
- 2進数の01001111
- 2進数の1001000010

## 実習で作ったプロジェクトを家に持って帰る1 (USBメモリを利用)

- マイドキュメント ▸ Visual Studio 2005 ▸ Projects を開く
- USBメモリをPCのUSBスロットに挿す
- USBメモリのフォルダを開いてファイルを表示
- Projects内のhelloをコピー
- USBメモリのフォルダ内で貼り付け
- USBメモリのフォルダを閉じる
- ハードウェアの安全な取り外し→USBメモリ
- USBメモリをUSBスロットから抜く

## 実習で作ったプロジェクトを家に持って帰る2 (圧縮→メール送信)

- マイドキュメント ▸ Visual Studio 2005 ▸ Projects を開く
- Helloフォルダを選んで圧縮
- 圧縮したファイルをメールに添付
- メールを自分に送る

## 実習

- ファイルをUSBにコピー
- ファイルを圧縮してメールに添付し自分のメールアドレスに送る

## 情報メディア館のプリンタでファイルを印刷(有料:1頁10円?)

- 用意するもの:USBメモリ，コピーカード
- USBメモリに書類(PDFファイル，テキストファイル，画像ファイル(jpgなど)をコピー
- USBメモリをプリンタのUSBスロット(ケーブル)に挿す
- タッチディスプレイの指示に従い，書類を選び，コピーカードを挿す
- 印刷
- コピーカードとUSBメモリを抜き取る

## 今回の宿題

- 実習3を実際に打ち込んで動作を確認
- プログラムを少し変更してエラーや表示の違いを観察する
- 自分の作成したプロジェクトのフォルダをUSB(などを使って)家に持って帰る
- Gold Finger Schoolの「日々の練習」を1回ずつTry!