

Cプログラミング演習1(再)

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
 - 即, 不合格の判定あり得る

はじめに

プログラミング言語も「言語」である

君は英語が話せるだろうか？

ウェブを使えば、言いたいことを英語に翻訳してくれる
それを話せば良い？

それでは「話せる」ことにはならない

はじめに(続)

ウェブを使えば知りたいことは出てくるけれども、「使いこなす」(きちんと話せる)ためには、以下のものが「自分の頭」の中に入っていないとダメ:

基本的な**単語**

文法

言語の**慣習**

段取り(どのように言えば伝わりやすいか)

Cも「言語」である

そして、Cを使いこなせることが君に期待されている
(少なくとも、『自分は理工系だ』と言えるためには...)

「ウェブ上にあるプログラム」や「他の人が書いたプログラム」をコピーしたのでは、**自分の頭を使ったことにはならない**

Cを使いこなせるようになるには、次が必要:

自分の頭で考え、手を使ってプログラムを打ち込み、
コンピュータを動かしてコンパイル（ビルド）し、
エラーが出たらその理由を考え、プログラムを実行し、
結果を確認する (変な結果が出たらもう一度考えなおし)

現状把握

何も見ないで次の答を紙に書いてください
できたら提出すること

課題1:

Hello, world!

と表示するCプログラムを作れ

え、簡単すぎる？

でも、これが基本なので、まずやってみましょう

(講義に出ていた人はやったばかりなのでできるはずですよ)

参考：Pythonという言語では
`print "Hello, world!"`
でできます

プログラミング言語によって
得意不得意があります
いろいろな言語が使えると
便利です

課題2.

課題1ができれば、

コメントをつけよう

コメントとはなんだっけ？

どう書くのだっけ？

できれば提出すること

Inetentionally blank page

ひとまずの答

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

```
int main(void)
```

```
{  
    printf("Hello, world!¥n");  
    return 0;  
}
```

課題3. 次の質問に答えてください。

(まだコンピュータを使わないで!)

1. 文字列の ¥n をとったら、どうなるでしょう？
2. ¥n の意味は？
3. return 0; をとって、コンパイルしたらどうなるでしょう？
4. return 0; は何のため？
5. printf を print にしてコンパイルしたらどうなるでしょう？
6. #include <stdio.h> を取り除くとどうなるでしょう？
7. なぜ#include <stdio.h> が必要なのでしょう？
8. int main() の意味は？ intはなぜ必要？
9. どのスペースが必要？どれが不要？

コメントの答

```
#include <stdio.h> // 標準入出力を可能にする
```

```
int main(void) // main関数  
{ // Hello, world!に続けて改行表示  
    printf("Hello, world!¥n");  
    return 0; // 正常終了  
}
```

課題3. 次の質問に教えてください。
(まだコンピュータを使わないで!)

1. 文字列の ¥n をとったら、どうなるでしょう？
2. ¥n の意味は？
3. return 0; をとって、コンパイルしたらどうなるでしょう？
4. return 0; は何のため？
5. printf を print にしてコンパイルしたらどうなるでしょう？
6. #include <stdio.h> を取り除くとどうなるでしょう？
7. なぜ#include <stdio.h> が必要なのでしょう？
8. int main() の意味は？ intはなぜ必要？
9. どのスペースが必要？どれが不要？

課題4. コンピュータを使いましょう

課題1のプログラムを「打ち込んで」 hello.c というファイルを作れ。

それをコンパイル（ビルド）し、

エラーが出たら修正し（どのようなエラーが出たかを記録すること）

できた実行ファイルを走らせ、

その結果を確かめよう

使うコンパイラ（プログラム開発環境）は自由です

Visual studio でも、bcc32でも cygwinのgccでも、好きなものを使ってください

それではやってみよう

```
#include <stdio.h>

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

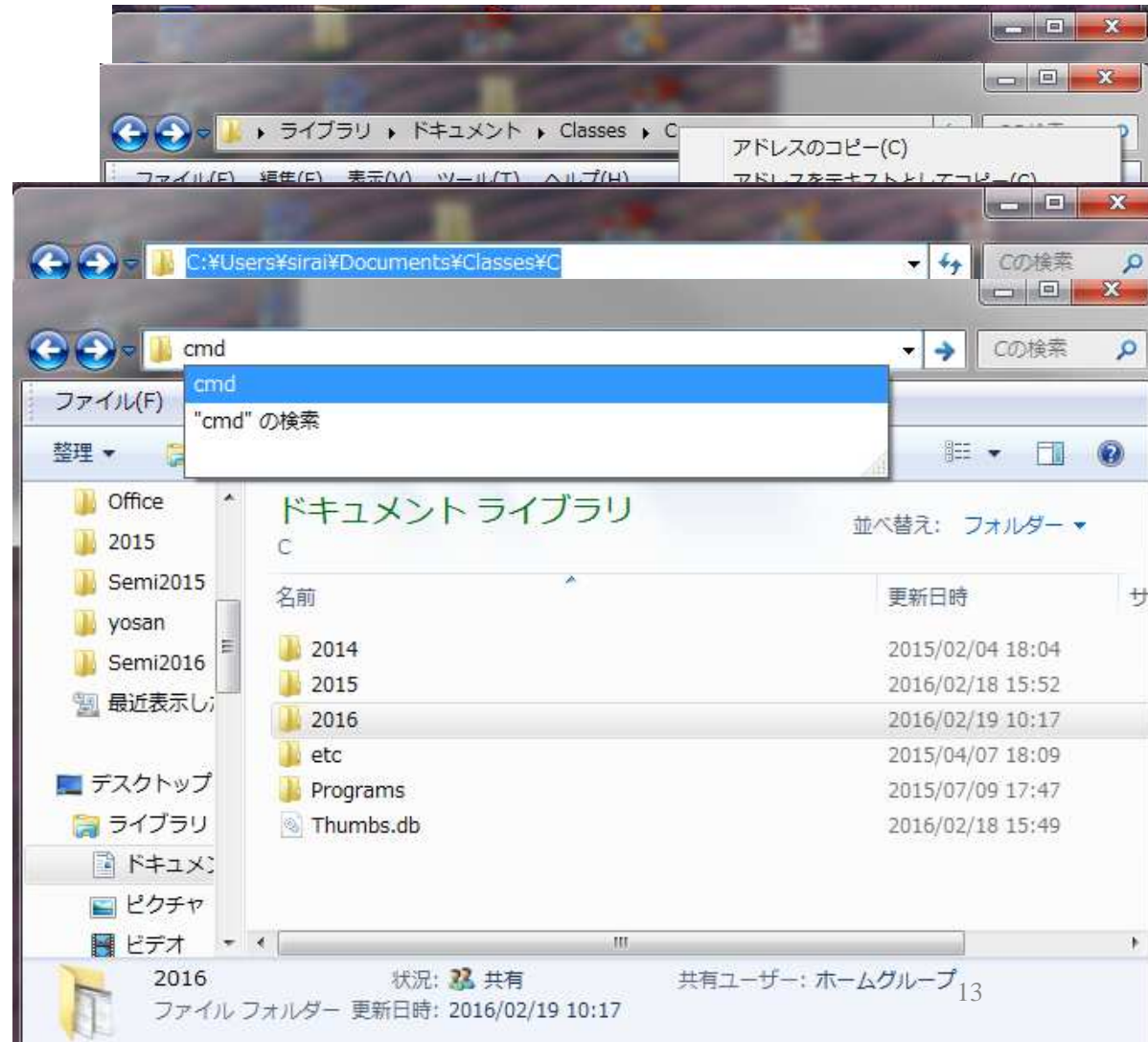
    return 0;
}
```

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ドライブのCEX
フォルダに 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

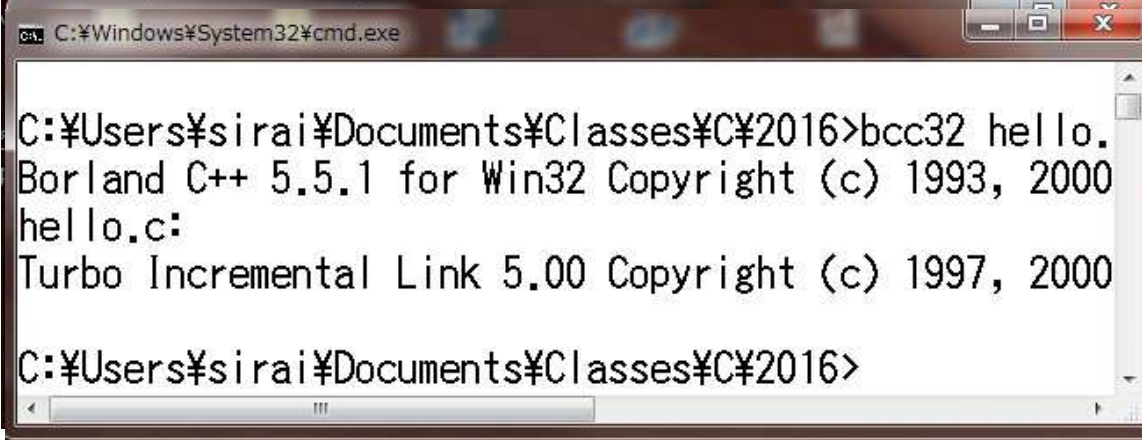
1. Hドライブを表示させる
(白井のフォルダーを例にする)
2. ファイルパスにマウスをもっていき、右クリック
3. 『アドレスの編集』を左クリック
4. 青地のラインにcmdと入力しエンターキーを押す



コマンドプロンプトにおいて

1. bcc32 hello.c

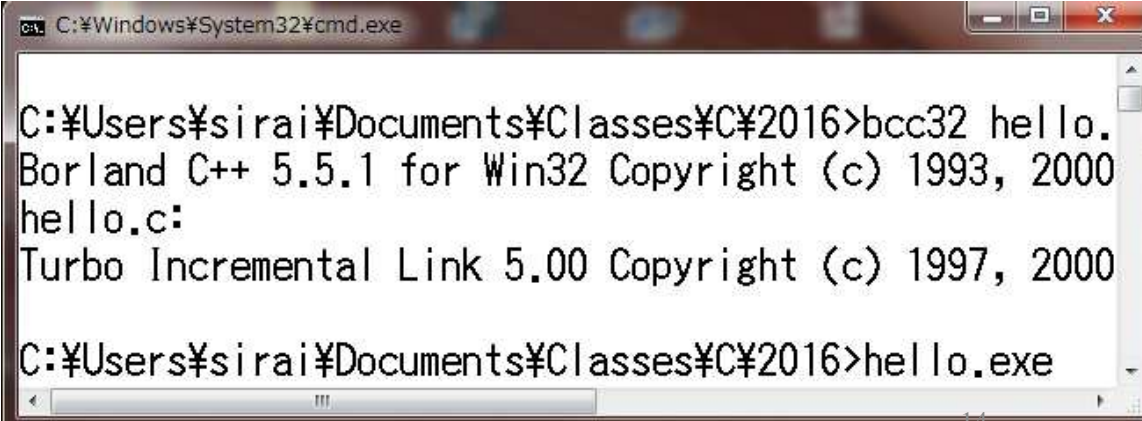
と入力(エンターキーを押す)



```
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
```

課題5

以下のように実行するCプログラムを書け(青字は入力):

お名前は? 白井

こんにちは、白井さん

ただし青字の部分を自分の名前（例：田中）に変えて

「こんにちは、田中さん」

と表示されるようにしよう

キーボードからの入力

課題 5

以下のように実行するCプログラムを書け(青字は入力):

お名前は? 白井

こんにちは、白井さん

このようなプログラムを書くには、

ユーザー：キーボードから「白井」と打ち込む

コンピュータ：打ち込まれた文字列を「記憶」し、それを出力に使う
ことが必要

ユーザーからの入力(続)

課題5

以下のように実行するCプログラムを書け(青字は入力):

お名前は? 白井

こんにちは、白井さん

コンピュータ: 打ち込まれた文字列を「記憶」し、それを出力に使うには、(1)「文字列を記憶するための変数宣言」、
(2)打ち込まれた文字列を記憶するための関数呼出
(3)記憶された文字列を表示するための関数呼出し
が必要

ユーザーからの入力 (続)

課題5

以下のように実行するCプログラムを書け(青字は入力):

お名前は? 白井

こんにちは、白井さん

コンピュータ：打ち込まれた文字列を「記憶」し、それを出力に使うには、(1)「文字列を記憶するための変数宣言」

```
char str[100]; // 100文字分記憶できる場所(配列) str
```

(2)打ち込まれた文字列を記憶するための関数呼出

```
scanf("%s", str) // 配列strに入力文字列を記憶
```

(3)記憶された文字列を表示するための関数呼出し

```
printf("こんにちは、 %s さん\n", str)
```

```
// 配列strに記憶された文字列を表示
```

プログラムができたら...

コンパイルし、実行してみましよう。

繰り返し(for)

ある一定の回数だけ、何らかのコードを実行する『繰り返し』にはfor文を用いる

例: 1から5まで、その3乗の数を表示する

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

```
int main(void)
```

```
{
```

```
    int i; // 繰り返しの回数を記録するための変数 i
```

```
    for (i=1; i <= 5; i++) // i=1, 2, 3, 4, 5 と5回繰り返す
```

```
    { // i*i*i (iの3乗) の値を表示する
```

```
        printf("%t%d^3 = %5d¥n", i, i*i*i);
```

```
    }
```

```
    return 0;
```

```
}
```

5回繰り返しの準備

繰り返される内容

繰り返しの例2 これは何をするものだろうか？

```
#include <stdio.h>
int main(void)
{
    int i, start, end;
    printf("Start Number? ");
    scanf("%d", &start);
    printf("End Number? ");
    scanf("%d", &end);
    for (i=start; i <= end; i++)
    {
        printf("%5d¥t%5d¥t%5d¥n", i, i*i, i*i*i);
    }
    return 0;
}
```

ヒント：数の読み込み：

`scanf("%d", &x);` // 変数xに入力した整数を記憶

課題 6

3個の整数を読み込み、その合計を表示するプログラムを書け。

実行例(青字は入力):

1 > 12

合計は 12

2 > 35

合計は 47

3 > -20

合計は 27

ヒント：

forを使って、3回以下を繰り返す：

- (1) 数を読み込む
- (2) それまでの合計を表示

プログラムができたら...

コンパイルし、実行してみましょう。

ヒント：

3回繰り返す：

- 1) 数を読み込む
- 2) それまでの合計を表示

繰り返し(while)

ある条件が成り立っている間、何らかのコードを実行する『繰り返し』はwhile 文を用いる

例: 1から5まで、その3乗の数を表示する

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

```
int main(void) {
```

```
    int i= 1; // 繰り返しの回数を記録するための変数 i
```

```
    while ( i <= 5 ) // i <=5 の間 繰り返す
```

```
{    // i*i*i (iの3乗) の値を表示する
```

```
    printf("¥t¥d^3 = ¥5d¥n", i, i*i*i);
```

```
    i++;
```

```
}
```

```
return 0;
```

```
}
```

5回繰り返しの準備

繰り返される内容

課題 7

0が入力されるまで、整数を読み込み、その合計を逐一表示するプログラムを書け。

実行例(青字は入力):

> 12

合計は 12

> 35

合計は 47

> -20

合計は 27

> 0

ヒント：

入力された数を記憶する変数を用意する
その初期値には0以外の値をセットする
whileを使って、入力された数が0以外なら
以下を繰り返す

(1) 数を読み込む

(2) それまでの合計を表示

プログラムができたら...

コンパイルし、実行してみましょう。

ヒント：

入力された数を記憶する変数を用意する
その初期値には0以外の値をセットする
whileを使って、入力された数が0以外なら
以下を繰り返す

- (1) 数を読み込む
- (2) それまでの合計を表示

課題 8

今日出てきた

printf

scanf

for

while

の機能（何をするものか）と特徴（どのような引数を取るか）を書け

宿題

課題 5 ~ 8 に対し

作業した内容

作成したファイル（ソースファイルのみ）

実行結果（エラーのあるなしも含めて）

をMANABOに報告する

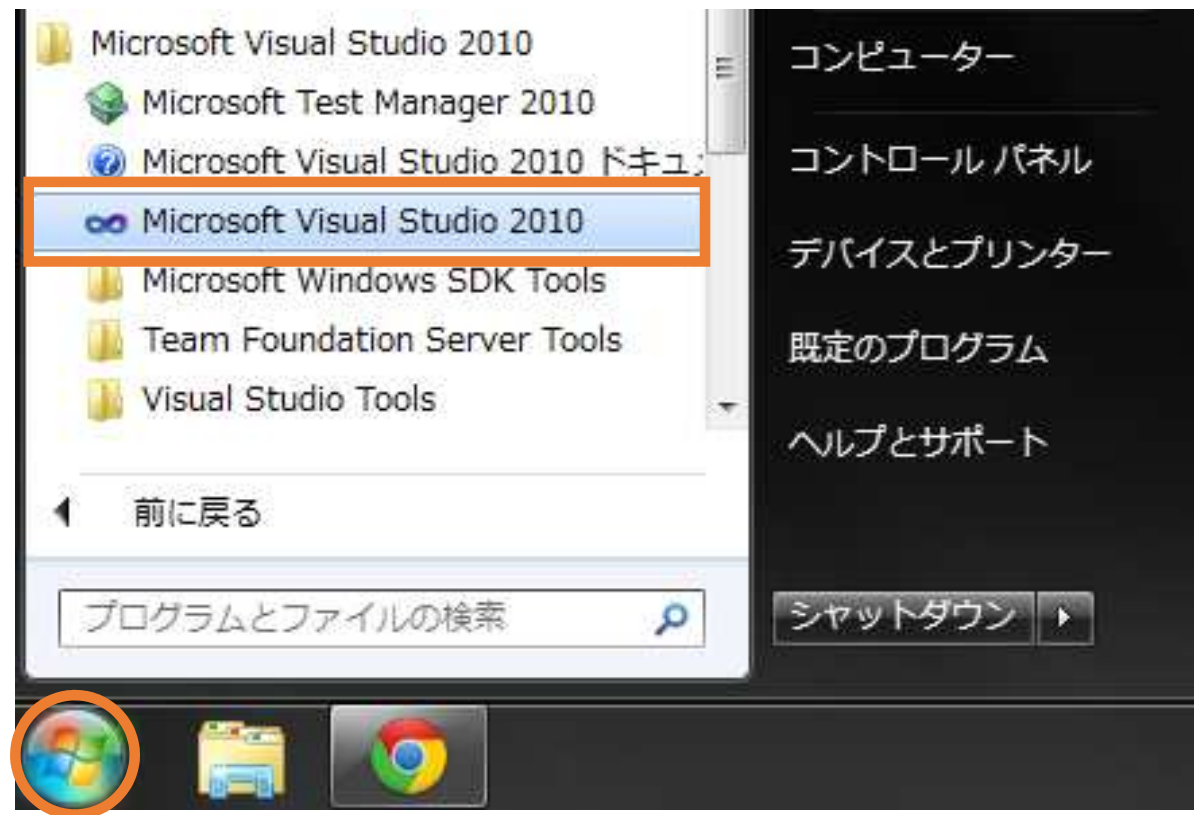
Inetentionally blank page

Borland Cによるコンパイル・実行

1. Tera Padや秀丸エディタでプログラムを書く
2. そのプログラムを適当なフォルダ（例: H:C2015）に置く---
ファイル名を test.c と仮定する
3. H:C2015フォルダを表示して Shift-右クリックで「コマンド
ウィンドウをここで開く」を選択
4. コマンドプロンプト画面で `bcc32 test.c` と入力
⇒ Borland Cコンパイラにより test.exe ができる
⇒ エラーメッセージが表示され、test.exe ができない場合は1へ
5. test.exe を入力するとプログラムが実行される

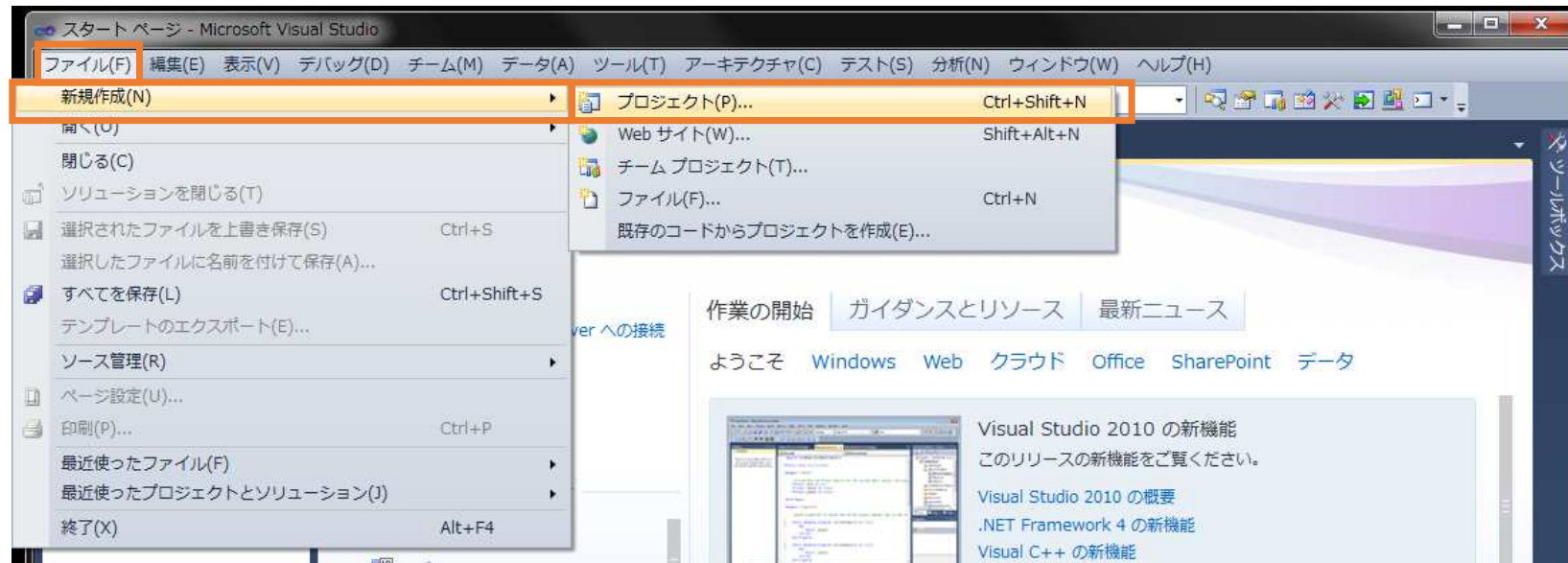
開発環境の準備

- スタートメニューから以下の枠内のメニューを探す



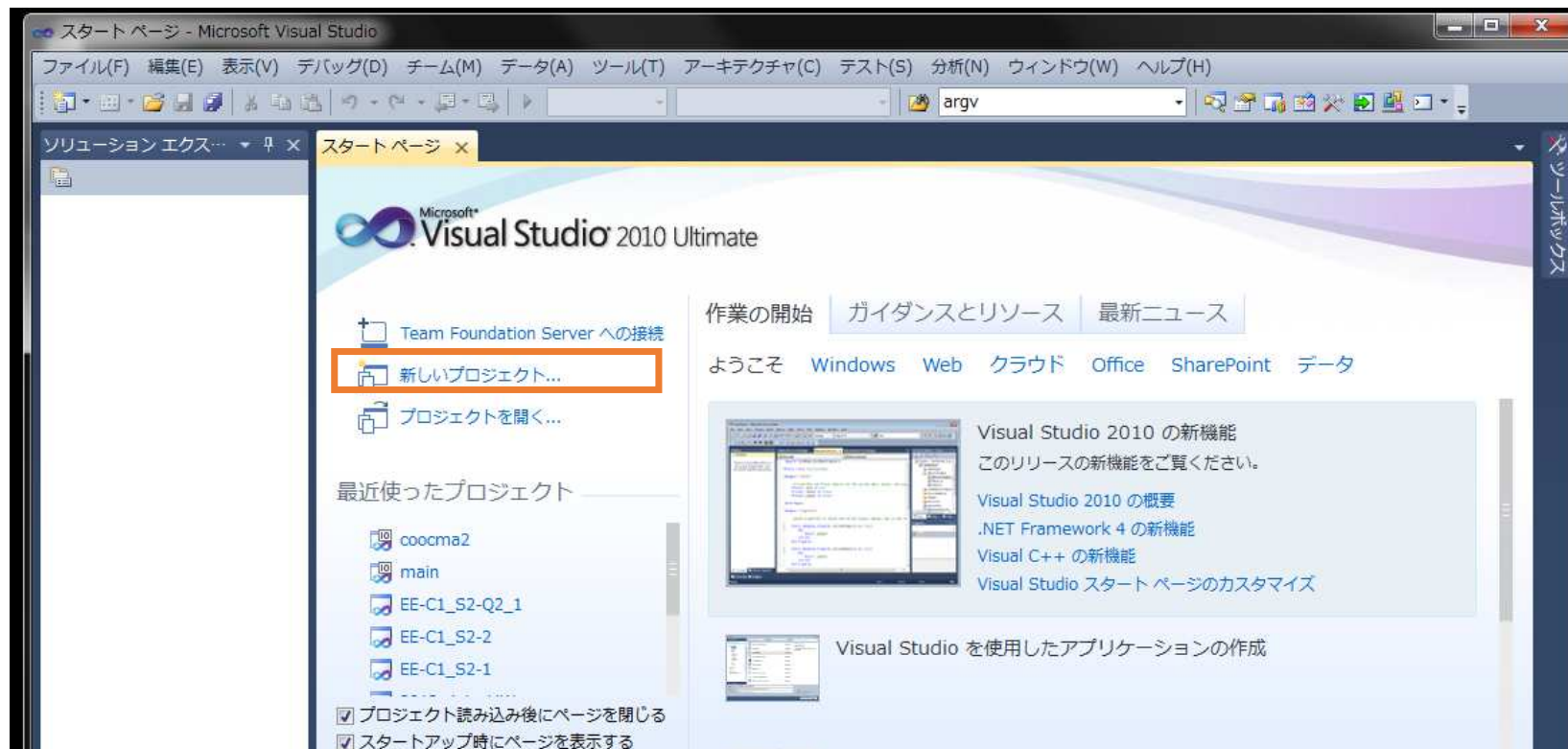
開発環境の準備

- プロジェクトの作成



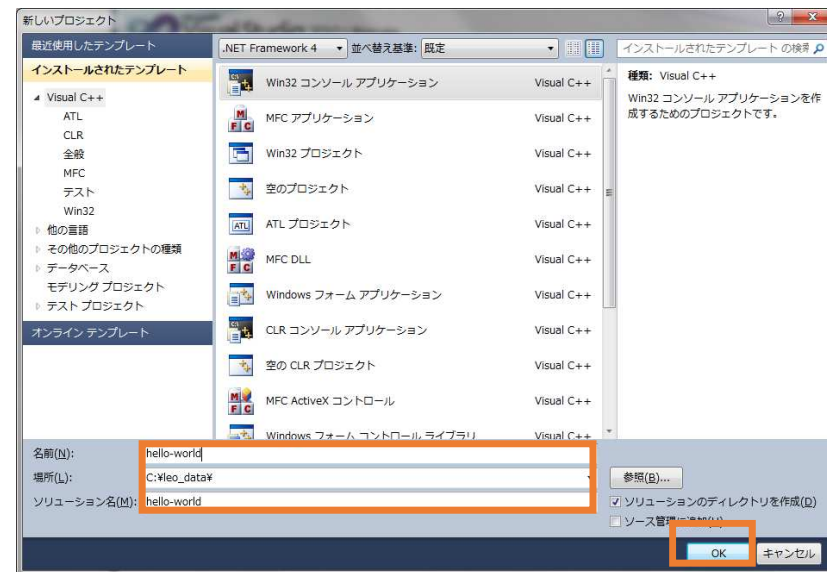
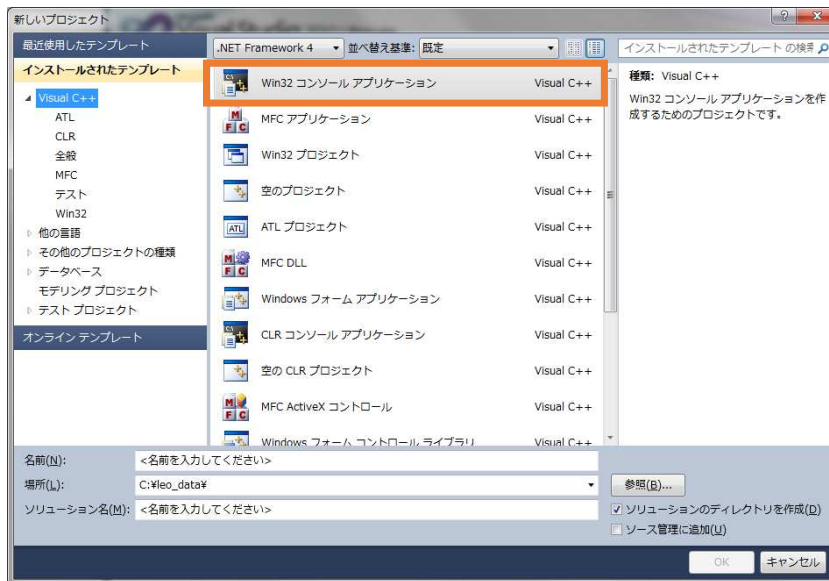
開発環境の準備

- プロジェクトの作成(別方法)



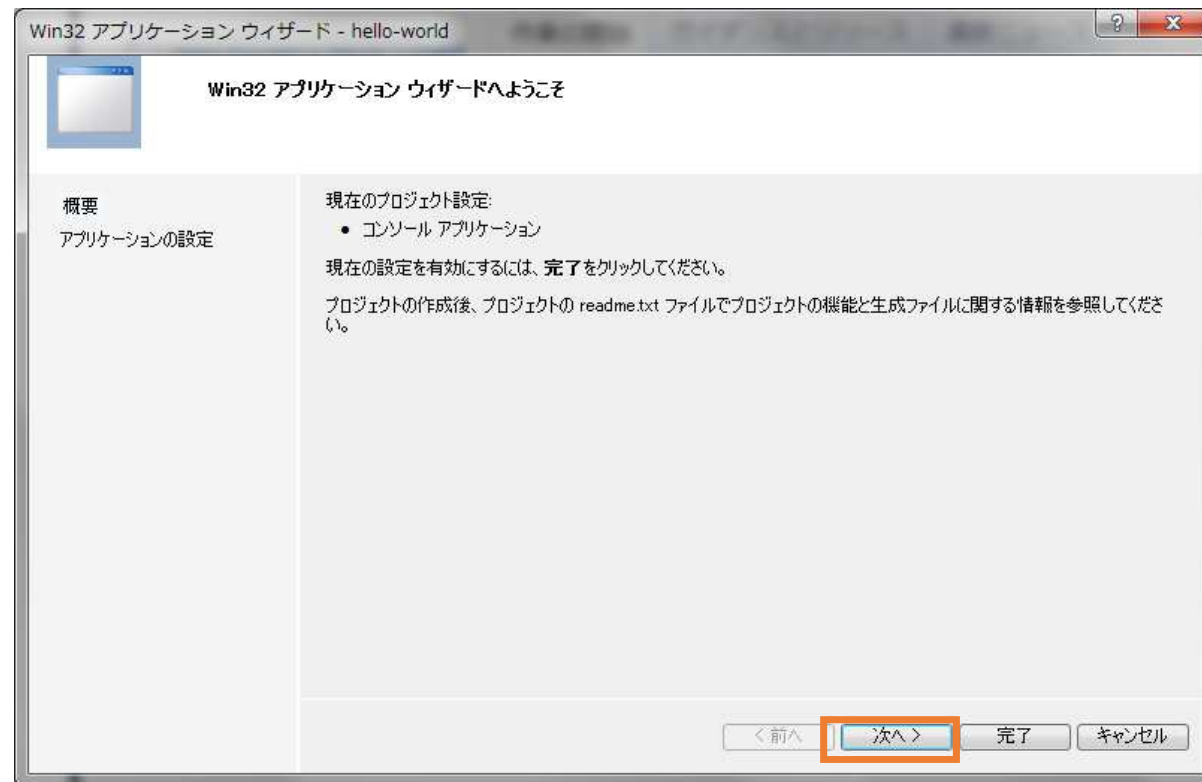
開発環境の準備

- 開発内容の選択
- プロジェクト名の記述



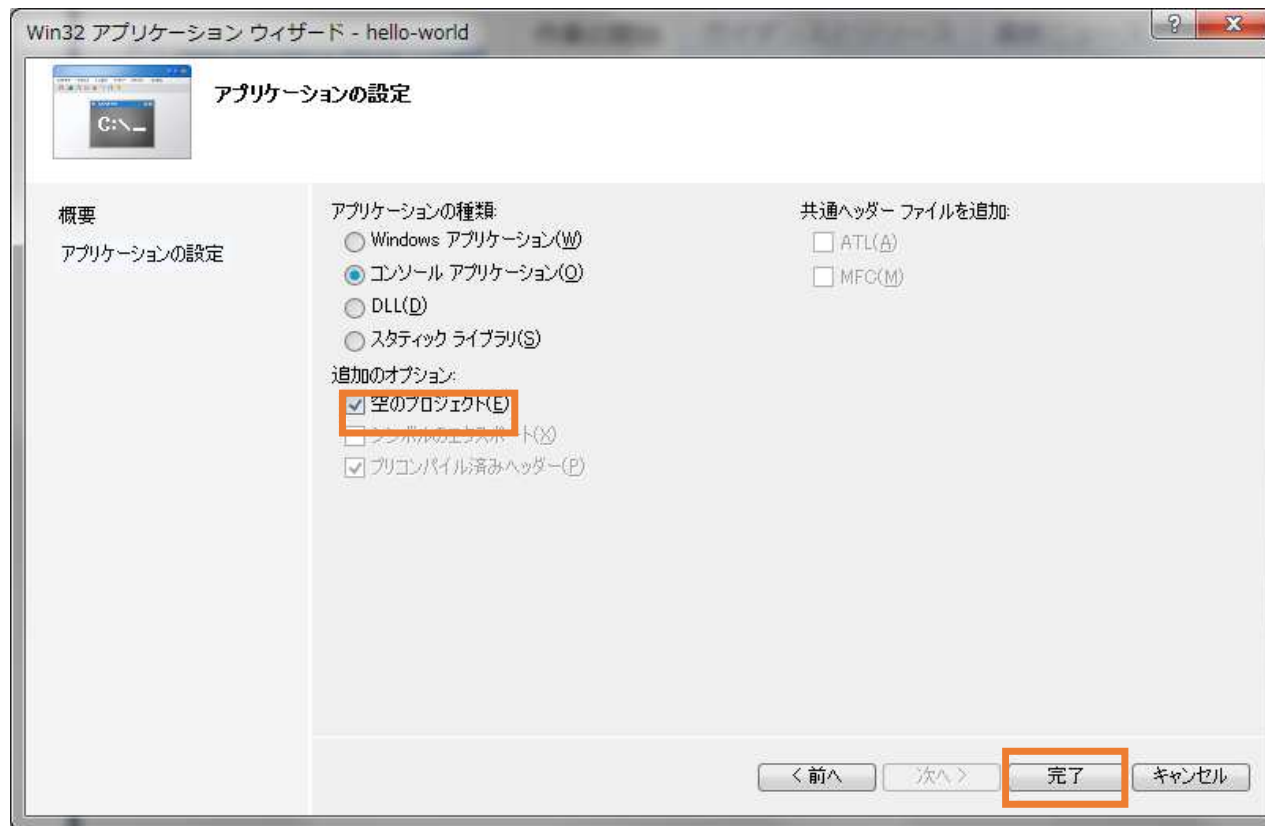
開発環境の準備

- ウィザードに従って設定



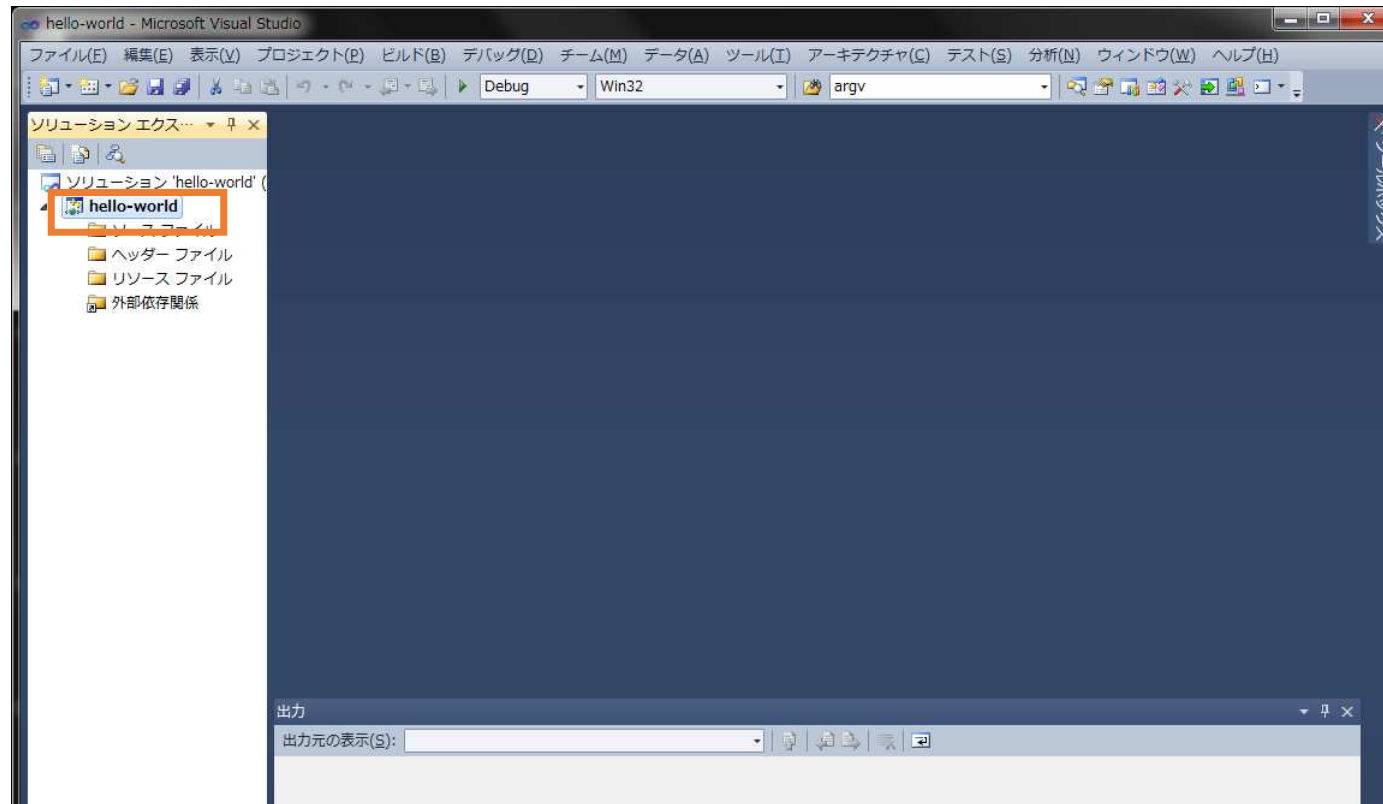
開発環境の準備

- 空のプロジェクトにチェックして完了



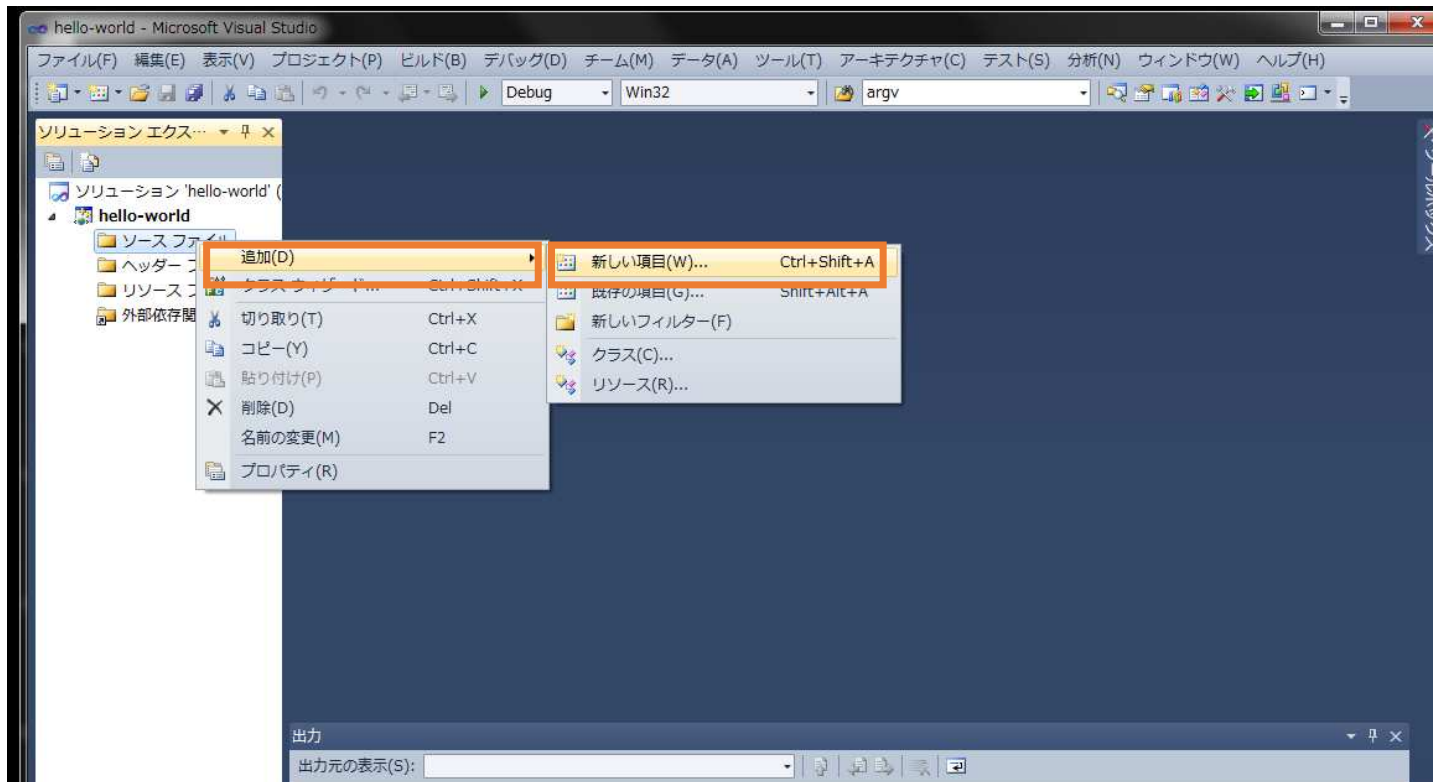
開発環境の準備

- 枠内アイコンを右クリック



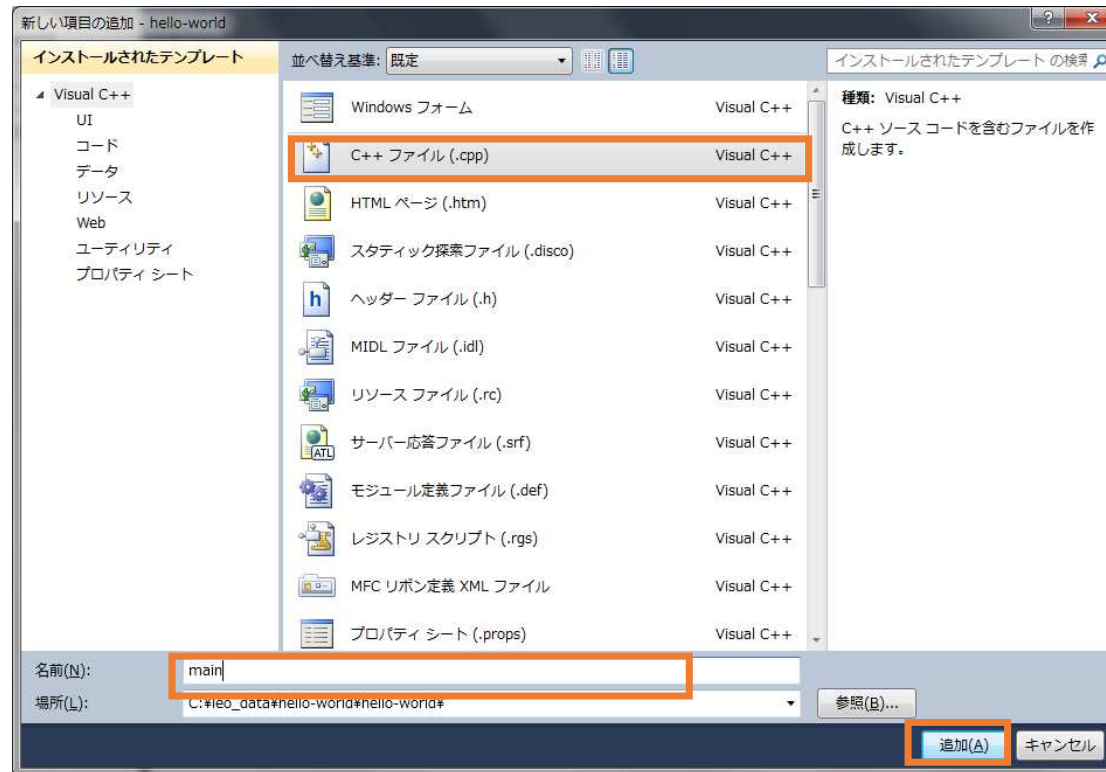
開発環境の準備

- ソースファイルの準備



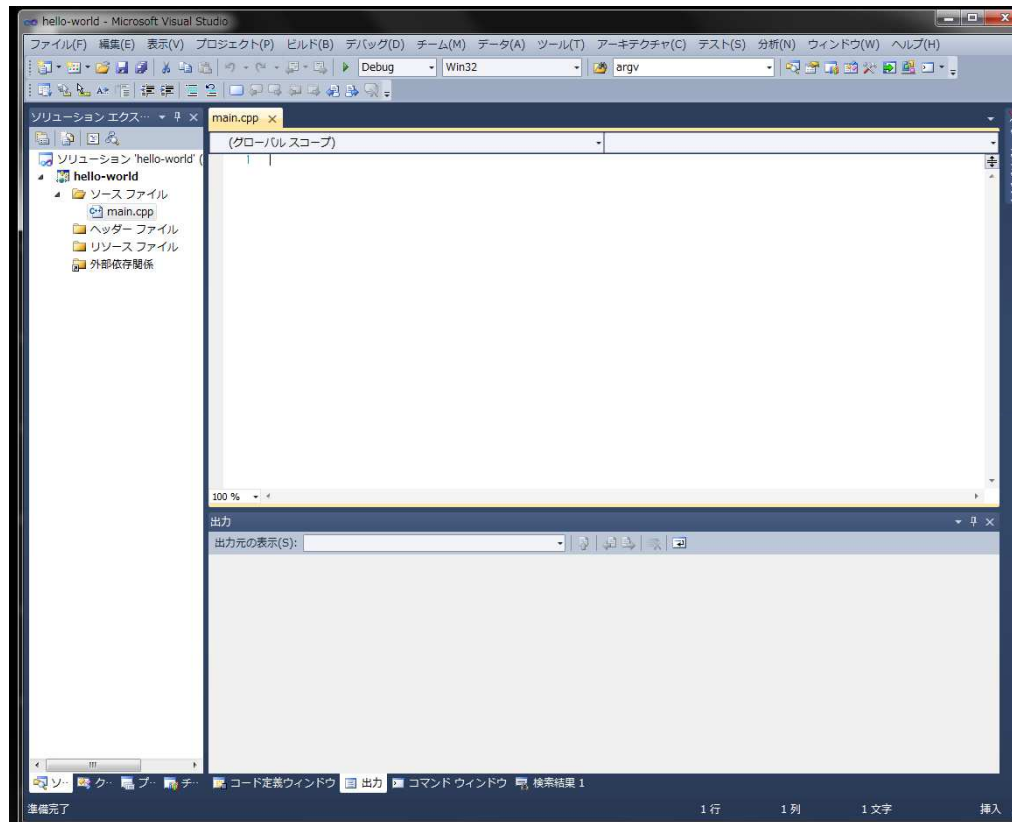
開発環境の準備

- ファイルの種類を選択，名前の決定



開発環境の準備

- 準備完了画面



開発環境の準備

- プロジェクトの保存

