

計算機プログラミング演習 ガイダンス（2021 年度前期）

場所： 対面授業の場合は工学部コンピュータ実習室(B207) ただし、教室を変更することもある。その場合は WebClass で通知する

1. 2021 年度の授業形式について（コロナウィルス対応）

本授業ではコロナ対策として、第1回と第2回は全員リモート（オンデマンド）となるので、資料を参考にしてプログラミングの環境構築を完了しておく。それ以降は、**コロナの状況に応じて対面授業にする**。その場合も通常の対面授業、対面授業とリモート授業（オンデマンド）を同時にすすめるハイブリッド形式の2つの形式を使い分ける可能性がある。ハイブリッド形式の場合は、受講生を2つのグループ（AとB）に分ける。例えば、Aグループが対面授業の週は、Bはリモート（オンデマンド）授業となる。次の対面授業はB、リモート（オンデマンド）授業はAとなる。このように対面をグループごとに交互に実施する。もしハイブリッド形式で行う場合は、WebClass 上クラス分けを提示するので注意しておくこと。**宮崎県内のコロナ感染の状況によっては完全にリモートにすることもある。コロナの状況に応じて臨機応変に行うので、WebClass は常に確認すること。**

2. 演習に際しての留意点

プログラミングは技術であり、プログラミングに関する概念の習得には実際の作業を伴う相当の経験と幅広い知識が必要である。一般にプログラミング言語を用いてプログラムを自由に作成できるようになるまでには多くの文献や資料、数ヶ月の時間が必要である。憶えるだけでは不十分で、技術として「身につける」のでなければ役に立たない。この点が他の科目と著しく異なる点であり、習得を困難にしている点である。しかし、**プログラミングはどの分野のエンジニアにも必須の技術になりつつある**。そのためには如何に多くのプログラミング経験をするかが重要である。失敗（エラー、バグ）を克服する経験は絶対に必要である。チャレンジする気持ちで演習に臨んでほしい。また、**C言語はそこから派生した言語も多いため、他の言語を学ぶうえでも役に立つ言語**であり。将来、Java や Python 等の他のコンピュータ言語に取り組むうえの基礎となる。

学生同士がプログラム作成について相談することはアクティブラーニングの観点から推奨するが、得られたアイデアはあくまでヒントとして**自分自身のプログラムを作成することが重要**である。プログラムが未完成であっても自分自身のプログラムの完成に向けて奮闘した姿勢が感じられれば評価（加点）する。**逆に安直なプログラムのコピーは厳禁である**。レポートで提出したプログラムがコピーであると判断される場合、**コピー先、コピー元を問わず、再提出あるいは採点を0点にすることがある**。また、**授業や演習で作成したファイルは全て保存しておくこと。前のファイルを上書きして消すということがないように（作成した課題のファイルが全てあるかチェックすることもある）、保存先も把握しておくこと！**

3. 「計算機プログラミング」の授業との違い

「計算機プログラミング」（木曜 3-4 時限：Thi Thi Zin 先生）の授業では主に「C 言語」の文法と基本的な使い方を習得する。演習ではそれを利用してプログラムを作成するための知識と経験を習得することが求められる。エラーやバグへの対処やプログラムを作る上での考え方など、より実際的な方法を学ぶ。

また、目的を達成するプログラムは必ずしも一通りとは限らない。作成が簡単なプログラムもあれば、実行速度が速いプログラムもあるし、多機能でいろいろな用途に使えるプログラムを作ることも可能である。プログラムの「価値」は多様である。より良いプログラムを作成するためにはいろいろなテクニックを知ることが重要であり、これも経験を通して習熟するしかない。

4. 教科書について

演習は配布資料に従って説明を行い、Visual Studio Community の操作を通じてプログラム作成を行う。教科書を指定しているが教科書に沿って演習を行うわけではない。それでも教科書は演習時間中に必ず手元に置いておくこと。副読本に近い使い方になるが、プログラム作成中は教科書を開いて（索引から）疑問点を解決することやヒントを探すことがたびたび生じる。これはプログラムに携わる仕事に就いてからも時折起こることなので、教科書は将来にわたって長く使うことになる。教科書は必ず購入して、卒業しても手放さないことを勧める。

5. 演習日程

演習は16回の予定である。Thi Thi Zin 先生の授業「計算機プログラミング」の進行状況によって変更が生じることに留意すること。定期試験に関しては基本的に行わない予定である。再試験は実施しない。

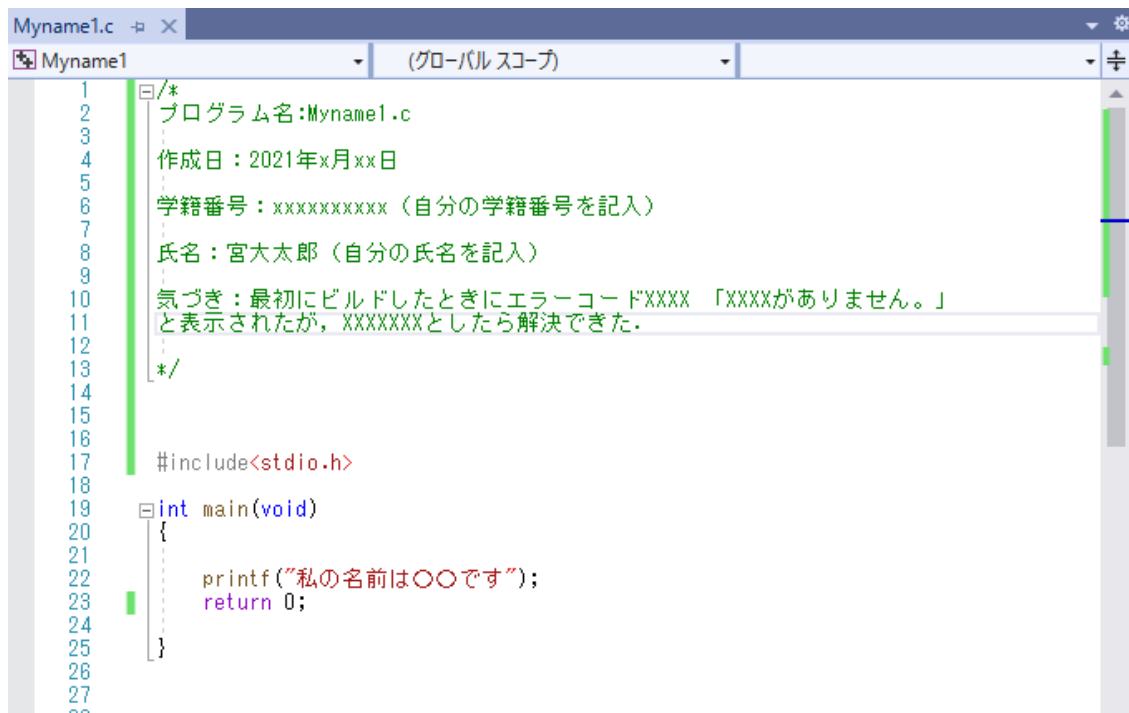
4年次生の就職活動での面接による欠席は2回で欠席1回にカウントする。ただし企業名と面接日時、面接場所を明記した報告書（A4）を速やかに提出すること。また面接による欠席は4回までとする。

6. 課題レポート提出要領

- ① ソースファイル（図1）、ビルド結果（図2）、実行結果（図5）の画面コピーを撮る（画面コピーの方法は色々あるがどの方法でもよい。やり方がわからない場合はネットで検索すると出てくる）。このとき、「プログラム名、作成日、学籍番号、氏名、気づき」をソースファイル冒頭にコメントで記入。コメントは/*と*/で囲んだ部分に記載すること。コメントとはプログラムで実行されないメモである。
- ② プログラム作成で気づいたこと（感想でもよい）や解決したエラーがあれば、「気づき」の箇所にコメントとして記載すること。したがって、正常な結果を提出するとともに、わざとエラーを起こして、その内容及び解決方法を「気づき」に記載することを勧める。**ここの記載は加点対象**としている。
- ③ どうしてもエラーが解決できない場合は、そのビルド結果（図3）だけでなくエラーメッセージ（図4）も提出すること。エラーメッセージはビルド結果の左下でエラー一覧を選択すると見ることができる。さらに、エラーの数字（図4ではC2143）やメッセージを調べて、「気づき」にどのようなときに生じるエラーかについて記載すること。
- ④ 撮った画面コピーをWORDに貼り付けた後にPDFとして保存。このときソースファイルのコードや結果の文字が小さくて見えないと評価できないので留意すること。
- ⑤ WebClassで提出するPDFファイルの名前は「kadaiX（Xは課題番号）_氏名（アルファベット）.pdf」とすること。例）kadai1_miyadai_taro.pdf

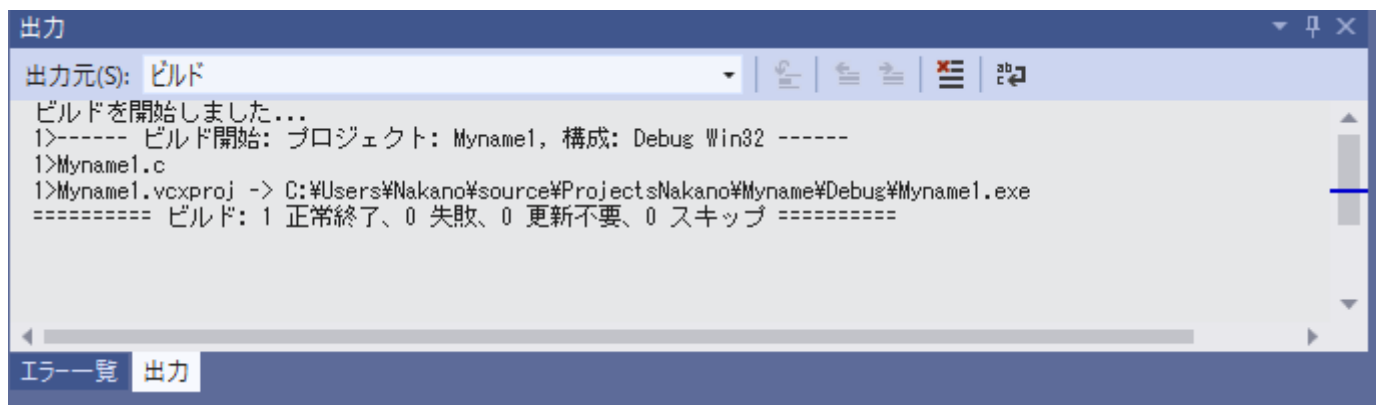
「Windows以外のOS（Mac, Linux）または開発環境を使いたい場合」

本講義の開発環境はVisual Studio 2019 Communityとしているが、どうしても他のOS（Mac, Linux）や開発環境（MacのXcode等）を使用したい場合は、不具合が生じた際に自分で対処できる人のみ認める。ただし、提出要領に準じた課題レポートを提出すること。



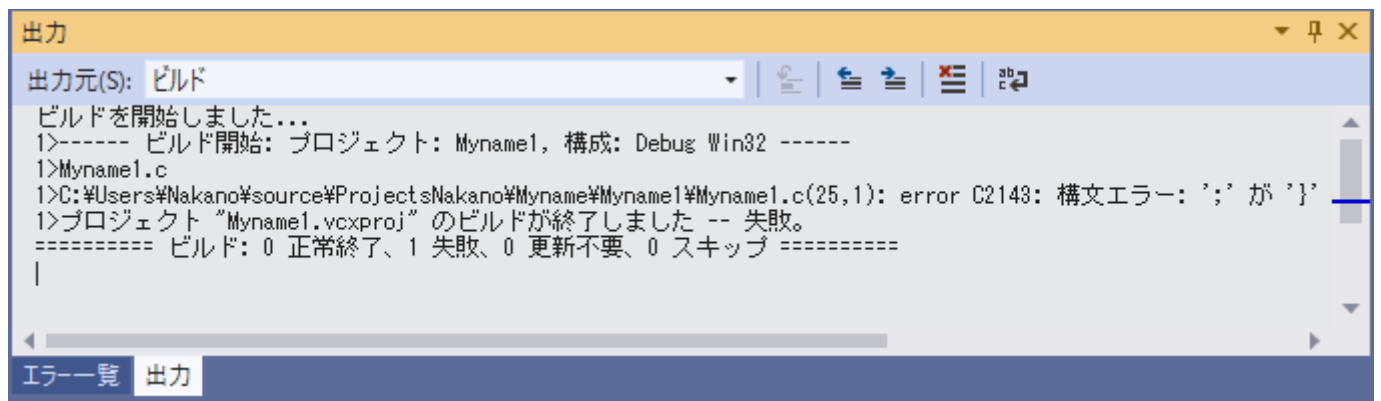
```
1  1
2  2  /*
3  3      プログラム名:Myname1.c
4  4      作成日：2021年x月xx日
5  5      学籍番号：xxxxxxxx (自分の学籍番号を記入)
6  6      氏名：宮大太郎 (自分の氏名を記入)
7  7      気づき：最初にビルドしたときにエラーコードXXXXX 「XXXXがありません。」
8  8      と表示されたが、XXXXXXとしたら解決できた。
9  9
10 10  */
11 11
12 12
13 13
14 14
15 15
16 16
17 17  #include<stdio.h>
18 18
19 19  int main(void)
20 20  {
21 21
22 22      printf("私の名前は○○です");
23 23      return 0;
24 24  }
25 25
26 26
27 27
28 28
```

図 1 ソースファイル例



```
出力
出力元(S): ビルド
ビルドを開始しました...
1>----- ビルド開始: プロジェクト: Myname1, 構成: Debug Win32 -----
1>Myname1.c
1>Myname1.vcxproj -> C:\Users\Nakano\source\Projects\Nakano\Myname\Debug\Myname1.exe
===== ビルド: 1 正常終了、0 失敗、0 更新不要、0 スキップ =====
```

図 2 ビルド結果例（成功）



```
出力
出力元(S): ビルド
ビルドを開始しました...
1>----- ビルド開始: プロジェクト: Myname1, 構成: Debug Win32 -----
1>Myname1.c
1>C:\Users\Nakano\source\Projects\Nakano\Myname\Myname1\Myname1.c(25,1): error C2143: 構文エラー: ';' が '}'
1>プロジェクト "Myname1.vcxproj" のビルドが終了しました -- 失敗。
===== ビルド: 0 正常終了、1 失敗、0 更新不要、0 スキップ =====
```

図 3 ビルド結果例（失敗）

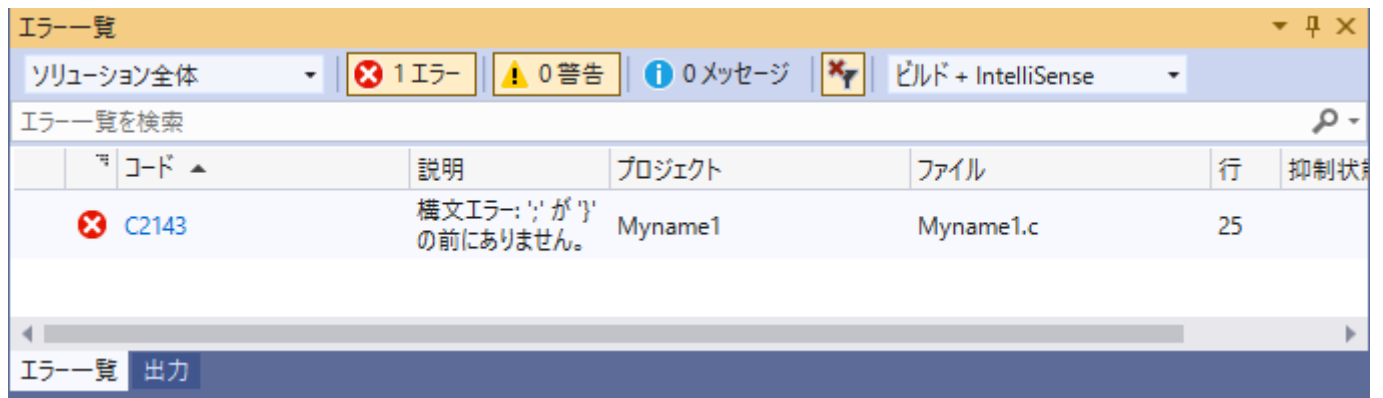


図 4 エラーメッセージ例（失敗）

```

C:\> コマンドプロンプト
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\> cd C:\Users\Nakano
C:\Users\Nakano> cd C:\Users\Nakano\source\Projects\Nakano\Myname\Debug
C:\Users\Nakano\source\Projects\Nakano\Myname\Debug> Myname1.exe
私の名前は〇〇です
C:\Users\Nakano\source\Projects\Nakano\Myname\Debug>
  
```

図 5 実行結果例

7. 課題レポート採点基準

- 下記は零点とする
 - 未提出
 - 締め切りを過ぎた課題の提出（ただし、やむを得ない理由がある場合を除く）
 - 安直にコピーしたプログラム（コピー先、コピー元も含めて）
 - 題意にそぐわないプログラム（課題に関係ないプログラム）

- 下記は減点対象とする
 - ソースプログラム または 実行結果 のどちらかの提出がない
 - プログラム 未完成 （コンパイルエラー，エラーメッセージなし）
 - プログラム 未完成 （コンパイルエラー，エラーメッセージ添付）
 - 実行結果が得られているが，結果が正しくない
 - 正しい実行結果が得られているが，場合によっては間違いが生じることがある
 - ソリューション名，プロジェクト名，プログラム名の使いまわし（以前作成したプログラムと同じ名前
→ 過去のプログラムが消去される）
 - ソリューション名，プロジェクト名，プログラム名のどれかが規定のまま変更していない（Project1,
Source.cpp プログラム名を見ても何のプログラムかわからない）
 - 字下げ（インデント）が適切でなく，プログラム構造が分かりづらい

8. 注意事項

誰かに教えて貰うのは良いが自分で作成すること．他の人や Web からの **コピーは絶対にしない** こと．
以前作成したプログラムに上書きをしない こと！！課題で作成したファイルは **全て残しておく** こと．