

Lecture 12

プログラミング演習Ⅰ その12

本日の演習の流れ

文字列と構造体について

course page

<http://amth.mind.meiji.ac.jp/courses/PE1/>

注意事項

リマインド

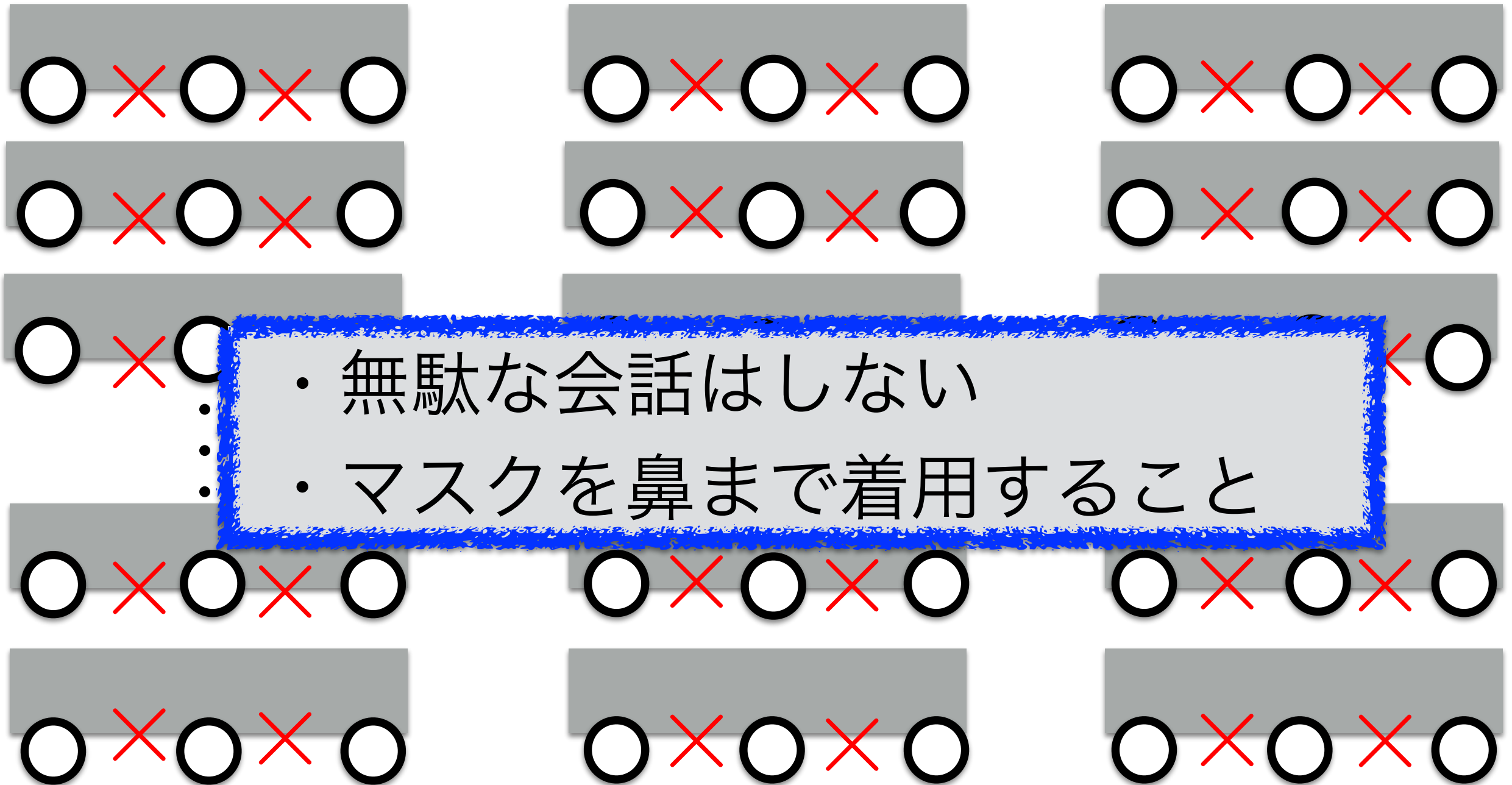
1. 授業のスライドはHPからダウンロードすることができますが、毎回の演習問題の回答例などは、自分で打ってください。

<http://amth.mind.meiji.ac.jp/courses/PE1/>

2. スライド以外の授業内容もあるので、ノートを取ること。
3. 授業中にスマートフォンやLineなどは使用禁止（演習問題の回答の写真撮影も不可！）
4. 偽造出席や出席名簿改竄（両者） = F.

教室での座席について

前



後

文字列 (strings) について

C言語には文字型 (char) はあるが、文字列型は存在しない。しかし、文字はいくつか集まって（文字列となって）初めて役に立つので、Cでは char の配列を文字列として使う習慣がある。

文字列の宣言の構文

```
char 文字列名 [文字数];
```

文字列をプログラム中使用したい場合は「“ ”」（ダブルクォーテーション）で囲む。

例) `char x[] = "penguin";`

注意 xの各要素には1文字ずつデータが格納される。

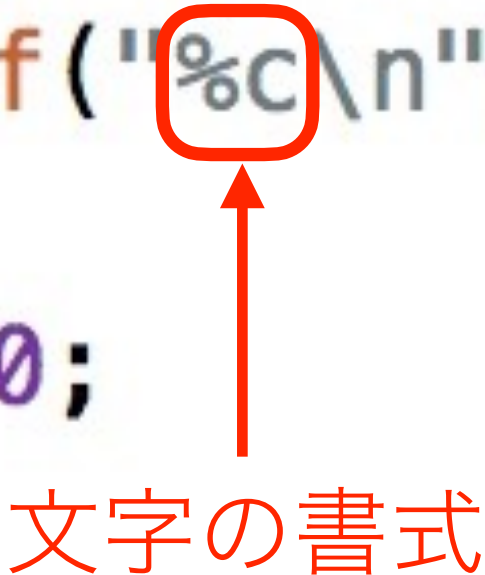
文字列の要素を変える場合は「' ’」（シングルクォーテーション）を使用する。

例) `x[0] = 'P';`

文字列

1 次のプログラムを実行してみよう.

```
1 #include <stdio.h>
2
3 int main(void){
4     char x[7]="penguin";
5     int i;
6     for(i=0;i<7;i++){
7         printf("%c\n",x[i]);
8     }
9     return 0;
10 }
```



文字の書式

文字列

2 (1)のプログラムと比べて、次のプログラムを実行してみよう.

```
1 #include <stdio.h>
2
3 int main(void){
4     char x[7]="penguin";
5     printf("%s\n",x);
6     return 0;
7 }
```

文字列全体を出力

s : string

文字列

- 3 文字列をキーボードから入力したい場合, `scanf` 関数を用いる.

以下のプログラムを実行してみよう.

```
1 #include <stdio.h>
2 int main(void){
3     char mojiretu[11];
4     scanf("%s",mojiretu); // scanf("%s",&mojiretu[0]); と同じ
5     printf("%s\n",mojiretu);
6     return 0;
7 }
```


構造体 (structures) について

変数の集まりである配列を使うと、多数のデータを取り扱う際に、大変便利であった。しかし、配列では同じデータ型の変数の集合体であった。これに対して、構造体は異なったデータ型の変数をまとめて取り扱うことができる。

構造体を使用するにはあらかじめ、構造体の構造を決め、次の形式で宣言しておかなければならない (**main()関数の前!**)。

struct タグ名{

変数宣言

...

};

structの最後に記述された名前を**タグ名 (tag)** といい、
構造体を構成している変数を**メンバ (member)** という。

注意

構造体宣言の最後に「;」がつくことに注意

構造体の宣言の構文

struct タグ名 構造体変数名 ;

構造体について

構造体変数の各メンバ (member) は次の形式で参照できる.

構造体変数名 . メンバ名

例を見てみよう.

4 a

```
1 #include <stdio.h>
2
3 struct meibo{
4     int num;
5     char name[30];
6     int age;
7 };
8
9 int main(void){
10     struct meibo student;
11     printf("student number: ");
12     scanf("%d",&student.num);
13     printf("student name: ");
14     scanf("%s",student.name);
15     printf("student age: ");
16     scanf("%d",&student.age);
17     printf("you entered the following:\n");
18     printf("%d\n",student.num);
19     printf("%s\n",student.name);
20     printf("%d\n",student.age);
21     return 0;
22 }
```

構造体 meibo の定義

構造体変数の宣言

メンバの初期化

構造体について

4b

```
1 #include <stdio.h>
2 #include <string.h>
3 struct book{
4     char title[30];
5     char author[30];
6     int num_pages;
7     int pub_year;
8 };
9
10 int main(void){
11     struct book favorite;
12     strcpy(favorite.title,"The old man and the sea");
13     strcpy(favorite.author,"Ernest Hemmingway");
14     favorite.num_pages = 108;
15     favorite.pub_year = 1952;
16     printf("%s\n",favorite.title);
17     return 0;
18 }
```

文字列を取り扱う関数

文字列の複写

構造体と関数

- 5 xy平面の点 (x,y) を考える. 点 $P(x_1, y_1)$ と $Q(x_2, y_2)$ のユークリッド距離は, 次の式で求められる.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

点 P, Q を構造体で表現し, ユークリッド距離を求める関数 `dist()` を定義せよ.

ヒント

```
double dist(struct point p, struct point q){  
    ...  
}
```

構造体と関数

- 6 自分で選んだ「物体（キャラクターなども可）」を表現する構造体を作成せよ。また、作った構造体を引数とする関数も作成せよ。

文字列と関数

- 7 文字列を前後逆転する関数を作成せよ。ただし、ポインタを使用すること。

`char str[7] = "penguin";` \implies `niugnep`

余談

In English, we say:

Write a function to reverse a string.