

# 11年度C言語講座

11/5/19(木) 講師:フェイス,オクウ,蟹男

# 前回の復習

- C言語についてのアウトライン
- 二進数C言語基礎の基礎及びVBの簡単な扱い方
  - “Hello World”プログラム
    - コンソールアプリケーション『ほぼ』最小構成
  - 四則演算
    - 『変数』、『代入演算子としての”=”』
  - 小数
    - 変数の『型』
  - 入出力基本
    - 『printf』、『scanf』の文法

# 条件分岐

- プログラムにおける重要な要素
  - これがないとプログラムは「何も判断できない」。
- コンピュータは0か1かしか判断できない。
  - 起こりうる全ての事象に対して、コードの段階で回答（真か偽かの二択）を用意しておく必要がある。
  - 実際問題全てというのは無理なので、  
「こちらが想定した事象に対する処理」  
「例外処理」  
を行う。

# if文

C1

```
#include <stdio.h>

int main(){
    int i;
    i = 10;
    if(i == 10){
        printf("i は 10 です。¥n");
    }
    else{
        printf("i は 10 ではありません。¥n");
    }

    return 0;
}
```

- 「～ならばこうする」
  - プログラムの根幹
  - if(条件1){処理1}  
else {処理2}の形で記述する。
- “==”は『等しい』
  - 通常のイコール

# 複数条件を指定するif文

C2

```
#include <stdio.h>

int main(){
    int i;
    printf("数値を入力してください。¥n->");
    scanf("%d", &i);
    if(i < 0){
        printf("i は 0 よりも小さいです。¥n");
    }
    else if(i >= 0 && I < 10){
        printf("i は 10 より小さい正の数です。¥n");
    }
    else{
        printf("i は 10 より大きいです。");
    }

    return 0;
}
```

- 1つのif文において複数の条件を指定することも可能
  - if(条件1){処理1}
  - else if(条件2){処理2}
  - else if(条件3){処理3}
  - ...
  - else{処理n}の形で記述する。

# switch文

C3

```
#include <stdio.h>

int main(void){
    int i;

    printf("簡単なジャンケンをしましょう。¥n 入力によって
           手を出します。¥n0->グー 1->チョキ 2->パー¥n");
    scanf("%d", &i);

    switch(i){
        case 0:
            printf("あなたはグーを出しました。¥n");
            break;
        case 1:
            printf("あなたはチョキを出しました。¥n");
            break;
        case 2:
            printf("あなたはパーを出しました。¥n");
            break;
        default:
            printf("不正な入力です。¥n");
            break;
    }

    return 0;
}
```

- 『条件を整数で指定するif文』

- switch(変数){

- case 整数:

- 処理1;

- break;**

- case 整数2:

- ...}

- のように記述する。

- 斜体部分に注意

- caseの後ろのコロン・break文

- 最後のdefaultは例外処理

# ……ゲーム？

- いくつかの問題点
  - そもそもジャンケンになっていない
    - 相手の出す手がないことが原因
  - 一度しか処理を行っていない
    - あいこの時は？
    - 続けてプレイしたいときは？
- 『乱数』と『ループ』
  - 条件分岐に続きゲームプログラムに重要な要素

# 乱数（擬似乱数）

C4

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

int main(void){
    srand(time());

    int i = rand()%3;
    printf("0 から 2 の間の数を表示します。¥n
    今回の数は%d です。", i);

    return 0;
}
```

- rand()で乱数を生成する（数値として扱える）。
- 生成されるのは0～（ある大きな数値）まで
  - 剰余演算子”%”を使い、任意の範囲に補正する。
    - 使い方は+や×と同じ

# ループ文 ①for文

C5

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

```
int main(void){  
    int i;  
    for(i = 0; i < 10; i = i + 1){  
        printf(“%d 回目の処理です。¥n”, i);  
    }  
  
    return 0;  
}
```

- for(初期値;条件;処理2)  
{処理1}

の形で記述する。

- ①初期値を指定し、  
②条件を満たす間、  
③処理1,2を繰り返す  
文。
- 主に、繰り返す回数が想定されているループや、ある数をカウントしていくループに適した記述。

# ループ文 ②while文

C6

```
#include <stdio.h>

int main(void){
    int i = 0;
    while(i < 10){
        printf(“%d 回目の処理です。¥n”, i);
        i = i + 1;
    }

    return 0;
}
```

- while(条件){処理}の形で記述する。
  - ①条件を満たす間
  - ②処理を繰り返す文。
- 主に繰り返す回数のわからないループ(半無限ループ)に用いる。
  - 記述を間違えると本当に無限ループになってしまい、プログラムがフリーズしてしまう。
- ちなみに2つのサンプルコードはほぼ同じ動作をする。

# 変数の種類

- int型
  - 整数を保持する。  
フォーマット指定子は”%d”
- float型
  - 小数を保持する。  
フォーマット指定子は”%f”
- char型
  - 文字を保持する。  
フォーマット指定子は”%c”

# 条件演算子

- 条件を指定するとき  
に用いる演算子
  - 優先度をつけたい時には()  
で囲む  
(数学と同様)。
- 使い方は今講義サンプ  
ルコード参照。
  - わからないところがあっ  
たら別途フェイスまで。

$> = (< =)$	左辺が右辺以上(以下)
$= =$	左辺と右辺が等しい
$! =$	左辺と右辺が等しくない
$\& \&$	かつ
$   $	または

# 課題

- ユーザーに正の整数の入力を求め、
  - ① 1からその数までの総和を求め、
  - ② それが3の倍数であるかを判定し、出力するプログラムを作成せよ。
  - 例) ユーザーが10を入力
    - 求めるのは  $1 + 2 + 3 + \dots + 10 = 55$  で、これは3の倍数ではない。
    - 出力例は、”55, これは3の倍数ではありません。”
- (非必須) switch文の項で紹介した「簡単なジャンケンプログラム」を完成させる方法を考え、可能ならば実装せよ。
  - 乱数を用いる。
  - forとwhile、どちらの文を使うのが適切か考えよ。