

第1回C++講座

1.C++(しーぷらすぷらす)の世界へようこそ

C++はCよりも機能が充実しています。プログラムの組みやすさもC++の方がいい気がします。さて、C言語講座が終わっていきなりC++なんてやってダイジョブなの?と思うかもしれませんが、文法はCと同じなのでまったく問題ありません。むしろ、CをやっつてすぐにC++をやることでCとどう違うのかを実感することが出来るでしょう。

また、C++はオブジェクト指向言語です。なので、オブジェクト指向に触れながらC++を学習していこうと思います。

2.変数宣言の自由

Cでは変数の宣言は関数の先頭でなければなりません。しかし、C++では変数の宣言はどこでも行っていいことになっています。どこで行ってもいいのですがその際には変数のスコープ(有効範囲)に気をつけてください。変数は宣言した行以降から使用できます。また、{}の範囲内でのみ有効となります。

サンプル1を見てイメージをつかんでください。

```
////////////////////////////////////  
/*  
C++講座サンプル1  
変数宣言の自由  
by y.t  
*/  
#include <stdio.h> /*入出力のヘッダーファイル*/  
/*メイン関数*/  
int main()  
{  
    int a = 10;           //今までどおり  
    printf("a = %d\n", a);  
    int b = 50;          //先頭でなくてもOK  
    printf("b = %d\n", b);  
    for (int i = 0; i < 10; i++) //こんなことも可能  
    {  
        int v = i * 6;  
        printf("v = %d\n", v);  
    }  
    //printf("v = %d\n", v); //vはスコープを外れている  
    printf("i = %d\n", i);  
    return 0;  
}
```

//

3.デフォルト引数

関数の引数は、関数を呼び出すたびに代入しなければなりませんでしたが、デフォルト引数を使うと関数に渡す値の初期値を設定することができます。サンプル2を見ながら使い方をやっていきましょう。

//

```
/*
```

```
C++講座サンプル2
```

```
デフォルト引数
```

```
by y.t
```

```
*/
```

```
#include <stdio.h> /*入出力のヘッダーファイル*/
```

```
void func1(int a = 6, int b = 3);           //デフォルト引数を含んだ関数の宣言
```

```
//void func2(int a = 5, int b);         こんな使い方は出来ない
```

```
/*メイン関数*/
```

```
int main()
```

```
{
```

```
    func1();           //デフォルト引数があるので引数の省略出来る
```

```
    func1(4);         //この場合は第2引数を省略
```

```
    func1(1,5);       //デフォルト値を使いたくないときは引数に値を入れる
```

```
    return 0;
```

```
}
```

```
void func1(int a, int b)
```

```
{
```

```
    int c = a * b;
```

```
    printf("%d * %d = %d\n", a, b, c);
```

```
}
```

//

関数を宣言する際に初期値を設定してやればいだけです。ただし、一度デフォルト引数を設定したらその関数の引数リストは、以降デフォルト引数しか設定できないようになっています。

今度は関数を呼び出してみましよう。デフォルト値をそのまま使う場合は引数を省略します。逆に任意の値を使いたいときは引数に代入をします。

4.関数のオーバーロード

関数名が同じでも、引数リストが異なる関数を作ることができます。作り方や使い方は実際にサンプル3をみてやっていきましょう。

//

```
/*
```

```
C++講座サンプル3
```

```
関数のオーバーロード
```

第1回C++講座 (By y.t)

by y.t

*/

```
#include <stdio.h> /*入出力のヘッダーファイル*/
```

```
int Max(int a, int b);
```

```
int Max(int a, int b, int c); //関数名は同じだが引数の数が違う
```

```
/*メイン関数*/
```

```
int main()
```

```
{
```

```
    int a = 5, b = 3, c = 7;
```

```
    printf("a = %d\nb = %d\nMax = %d\n\n", a, b, Max(a, b));
```

```
    printf("a = %d\nb = %d\nc = %d\nMax = %d\n", a, b, c, Max(a, b, c));
```

```
    return 0;
```

```
}
```

```
int Max(int a, int b)
```

```
{
```

```
    if (a > b) return a;
```

```
    else return b;
```

```
}
```

```
int Max(int a, int b, int c)
```

```
{
```

```
    int max = a;
```

```
    max = max > b ? max : b;
```

```
    max = max > c ? max : c;
```

```
    return max;
```

```
}
```

```
////////////////////////////////////////////////////////////////////
```

関数の宣言を見てください。関数名は同じですが、引数の数が違います。このように、処理内容は同じでも引数の型や数の違いで動作が異なるような状況で使えます。

関数の作り方自体はまったく変わりません。

5.bool 型

C++には `bool` 型というデータ型が存在します。このデータ型には `true` (真)か `false` (偽)の2つしか入りません。

使い方のイメージはサンプル 4 でも見てください。

```
////////////////////////////////////////////////////////////////////
```

```
/*
```

```
C++講座サンプル 4
```

```
bool 型
```

```
by y.t
```

```
*/
```

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

