

#### 4. 関数 - ローカル変数とグローバル変数の違い, スタティック変数の使い方 -

グローバル変数はプログラムのどこからでも参照・更新することができる変数である。しかし、いつでも、どこからでも変更することができるため、バグの原因になりやすい。したがって、関数間でのデータの受け渡しに使用することは望ましくない。そのような場合はできるだけローカル変数と引数などを使うことが望ましい。どうしても使いたいときはグローバル変数だとわかるような変数名や長い変数名を付けることを推奨する。グローバル変数はプログラム全体で共有する特別なデータに使用されることが一般的である。

/\* 例題 4-1 絶対値を求める \*/

```
1: #include <stdio.h>
2:
3: int abs_value(int l_num)
4: {
5:     if(l_num >= 0) {
6:         return l_num;
7:     } else{
8:         return -l_num;
9:     }
10: }
11:
12: int main (void)
13: {
14:     int num;
15:     scanf("%d", &num);
16:     printf("%d\n", abs_value(num));
17:     return 0;
18: }
```

変数 l\_num のスコープ (利用できる範囲)

/\*例題 4-2 例題 4-1 でグローバル変数を利用\*/

```
1: #include <stdio.h>
2:
3: /* global 変数 */
4: int g_num;
5:
6: void g_abs_value(void)
7: {
8:     if(g_num < 0){
9:         g_num = -g_num;
10:    }
11: }
12:
13: int main (void)
14: {
15:     scanf("%d", &g_num);
16:     g_abs_value();
17:     printf("%d\n", g_num);
18:     return 0;
19: }
```

変数 g\_num のスコープ (利用できる範囲)

global 変数 g\_num を絶対値に変換する

/\*例題 4-3 値を入力するたびにそれまでの合計を返す関数\*/

```
1: #include <stdio.h>
2:
3: int goukei(int a)
4: {
5:     /* static 変数 */
6:     static int s_num = 0;
7:
8:     s_num += a;
9:     return s_num;
10: }
11:
12: int main(void)
13: {
14:     int i, a;
15:
16:     for(i = 1; i < 10; i++){
17:         printf("整数を入力してください\n");
18:         scanf("%d",&a);
19:         printf("今までの合計は%d\n",goukei(a));
20:     }
21:     return 0;
22: }
```

変数 s\_num のスコープ

static 変数とは、変数が宣言されたブロック(つまり{ }で囲まれた部分のこと)が終了しても値が保持される変数である。他のブロックから参照/変更されないという利点がある。関数に以前の値を記憶させたい場合に用いられる

実は、static 変数は宣言の時に初期化子が無ければ 0 で初期化されるので、わざわざ 0 で初期化する必要はない。

static 変数の s\_num にはこれまでの合計の値が保持されている。これまでの合計の値に引数 a の値を足している。

1. 例題4-3の「static int s\_num」のstaticを消した場合, どのような結果になるか実行して確かめよ。
2. 例題4-3を基にして, 呼び出すとそれまでの平均値を返す関数 heikin() を作成し, プログラムを完成せよ. 値の入力回数は10回とする。
3. 二つの整数のうち小さい値を返す関数を定義し, プログラムを作成する。以下の設問ごとにプログラムを作成しなさい。
  - (a). 引数と戻り値を利用した関数を定義し, プログラムを作成せよ。
  - (b). グローバル変数を利用した関数を定義し, プログラムを作成せよ。
  - (c). (a)の関数を修正せずにそのまま利用することにより, 3つの整数値から最小値を出力するプログラムを作成せよ。

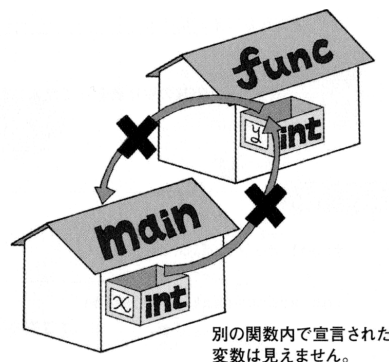
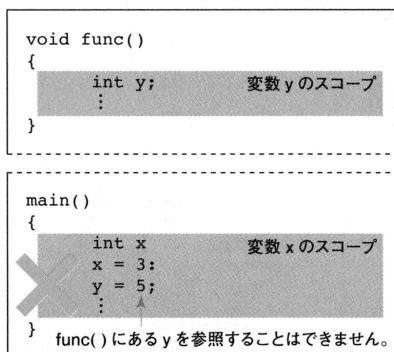
#### 第4章「関数 – ローカル変数とグローバル変数の違い, スタティック変数の使い方」のチェック項目

##### 変数の有効範囲について

- ☐ ローカル変数とグローバル変数で変数の有効範囲が異なることがわかった
- ☐ ローカル変数の有効範囲は宣言されたブロック内であることを理解した
- ☐ グローバル変数の有効範囲はプログラム全体であることを理解した

##### 変数の記憶寿命について

- ☐ ローカル変数とグローバル変数で変数の記憶寿命が異なることがわかった
- ☐ ローカル変数の記憶寿命は宣言されたブロック内の処理が終了するまでであることを理解した
- ☐ グローバル変数の記憶寿命はプログラムが終了するまでであることを理解した
- ☐ ローカル変数の宣言時に型名の前に「static」をつけることでグローバル変数と同じ記憶寿命に変更できることを理解した



##### グローバル変数のスコープ

