変数のスコープ

山本昌志*

2006 年 11 月 10 日

概 要
変数のスコープを表すローカル変数とグローバル変数について学習する．

1 前回の復習と本日の学習内容

1.1 復習

先週は，さまざまなユーザー定義関数を学習した．その内容は，以下のようなものであった．

- ファイルに計算データを書き込む方法を学習した．以下のように記述すれば，データをファイルに書き込むことができる．
  1. **FILE *out;**　　ファイルの情報を格納する変数宣言．`*out` がファイルの情報を格納する変数である．`out` は `fugafuga` のようにどんな名前でも良い．しかし，先頭のアスタリスク (*) は必須である．
  2. **out=fopen(”data.txt”,”w”);**　　書き込み用にファイルを開く．`data.txt` がファイル名で，`w` が書き込み用にファイルを開くことを示している．ファイル名は適当に変えてもよい．`fopen()` 関数の戻り値は，ファイル情報を格納する変数に代入する．
  3. **fprintf(out, ”%f\t%f\n”, x, y);**　　ファイルにデータを書き込む．ファイルに書き込む `fprintf()` 関数は，ディスプレイに表示する `printf()` 関数とそっくりである．書き込むファイルを指定するファイル情報の変数を最初に書く—ことが異なる．
  4. **fclose(out);**　　ファイルを閉じる．使い終わったファイルは閉じなくてはならない．
- 引数や戻り値の無い関数を宣言するときには，`void` と型宣言をする．

1.2 本日の学習内容

本日の学習範囲は教科書 [1] の p.189–202 である．教科書とは別に，プリントに沿って先週よりは複雑，そして役に立つ関数の作成を試みる．本日の学習のゴールは以下のとおりである．

- ローカル変数とグローバル変数の違いが分かり，変数のスコープと言う考え方を理解できる．
- グローバル変数の使い方が分かる．

*独立行政法人　秋田工業高等専門学校　電気情報工学科

2　変数のスコープ

変数はデータを記憶するためのもの—記憶領域に名前をつけたもの—である．それには，いろいろ便利な仕組みがある．それらの仕組みは，(1) 大規模なプログラムを効率的に記述，(2) メモリー—記憶領域—を節約，(3) 高速に動作させるためにある．諸君が作る小さなプログラムであればあまり恩恵はないが，将来必ず役に立つときがくる．

2.1　宣言の場所と有効範囲

変数は，宣言する場所によって有効範囲が異なる．この有効範囲のことをこれを変数のスコープ (scope: 範囲) という．有効範囲というのは，プログラム中で変数が使える場所のことである．具体的には，その変数を使って計算したり，表示することができる範囲で

```
printf("%f", hogehoge);
```

のように書いてもエラーにならない範囲である．

変数を宣言する場所，次の 3 箇所と考えてよい．

- 全ての関数の外側，プログラムの先頭付近で宣言した変数は全ての関数で有効である．これを**グローバル**変数と呼ぶ．
- 関数の先頭で宣言した変数は，その関数内のみで有効である．これを，**ローカル変数**と呼ぶ．また，関数の仮引数もローカル変数である．
- コードブロック—{　} で囲まれた部分[1]—の先頭で宣言した変数は，そのブロック内のみで有効になる．これもローカル変数のひとつであるが，ここでは**ブロック内宣言の変数**と呼ぶことにする．

この 3 つの変数宣言の場所とスコープの関係をを図 1 に示す．これをしっかり理解せよ．

これが理解できたならば，図 1 のプログラムの実行結果が，以下のようになることが分かるであろう．自分でこのプログラムの動作を追ってみよ．

実行結果

```
fuga at main = 222
hoge at main = 111
foo at test = 333
foo at test = 111
bar at for loop = 444
bar at for loop = 444
```

[1] `if` 文やループの時使った．

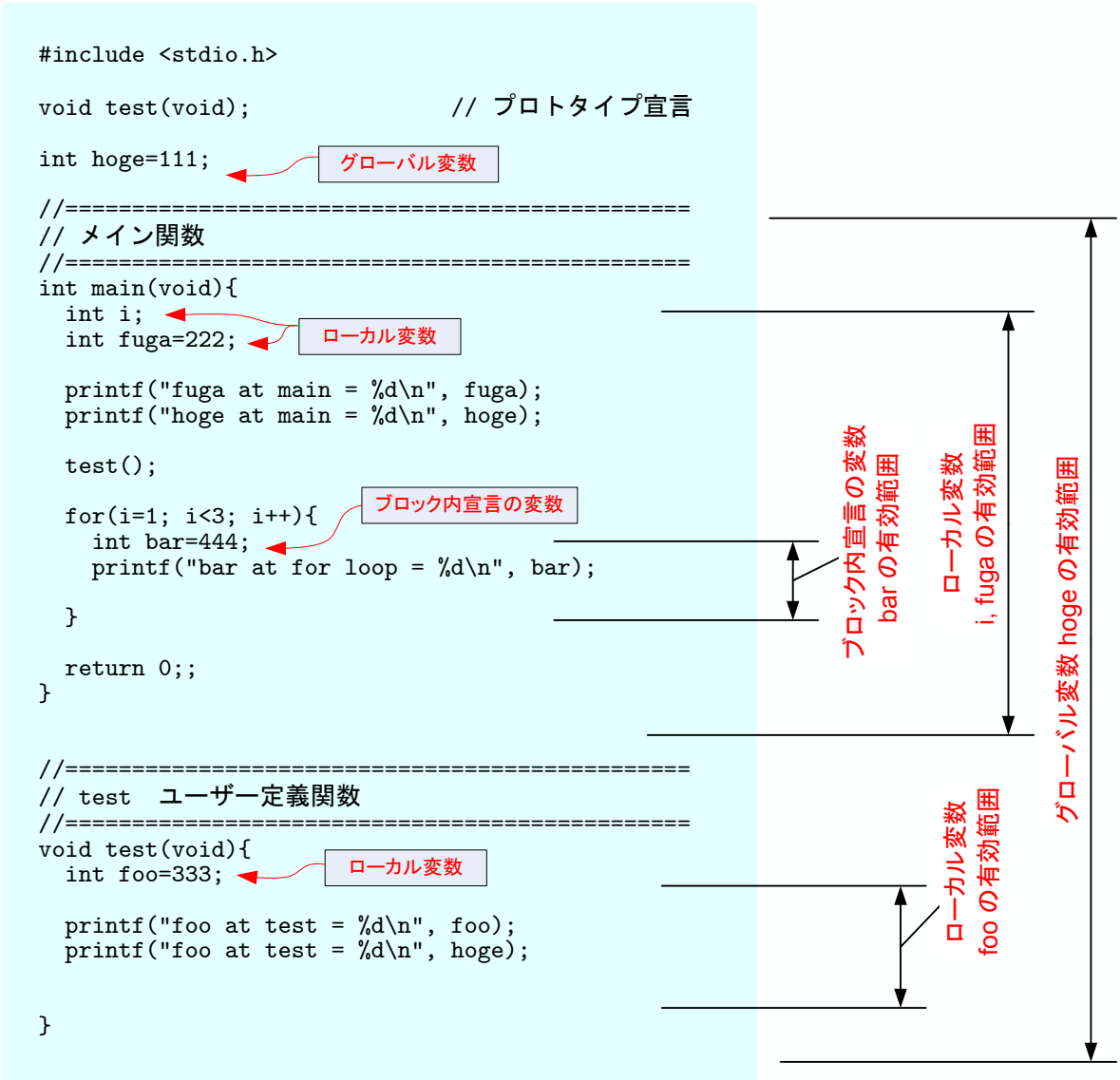

図 1: 変数のスコープ (有効範囲)

2.2 優先順位

どのような理由で，変数は宣言する場所でスコープが異なるのであろうか?．ちょっと考えると，グローバル変数だけでよいように思える．小さなプログラムであれば，グローバル変数だけでよい．実際，グローバル変数しかないプログラミング言語もある．このようなプログラミング言語は，大きなプログラムを作

ることは不可能で，ちょっとした小さなプログラム専用である．なぜならば，グローバル変数だけだと，同じ変数名を使うことができず，変数名の命名に大変苦労する．C 言語のように宣言する場所で，変数のスコープが異なると，同じ変数名を使うことができる．メイン関数で「hogehoge」と言う変数を使っていても，他のユーザー定義関数などで「hogehoge」を使っても，異なった物として取り扱ってくれる．これは，便利な機能で複数のプログラマーがそれぞれ関数を作成してひとつのプログラムを作る場合，各人が勝手な変数名を使うことができる．最初の疑問「変数宣言の場所でスコープが異なる理由?」の答えは，同じ変数名を使えることにするためで，それにより大規模なプログラムが開発できるようにしている．

それでは,「グローバル変数とローカル変数で同じ変数名を使った，どちらが実際使われるのか?」ということになる．そのため，同じ変数名には優先順位がある．優先度の高い順に並べると，(1) ブロック内宣言の変数，(2) ローカル変数，(3) グローバル変数となる．これを理解するために，同じ変数名を使ったプログラムをリスト 1 に示す．

リスト 1: 同じ変数名を使ったプログラム例

```
1  #include <stdio.h>
2
3  void test(void);                  // プロトタイプ宣言
4
5  int hoge=999;
6
7  //=============================================
8  // メイン関数
9  //=============================================
10 int main(void){
11   int i;
12
13   printf("hoge at main = %d\n", hoge);
14
15   test();
16
17   for(i=1; i<2; i++){
18     int hoge=888;
19     printf("hoge at for loop = %d\n", hoge);
20
21   }
22
23   return 0;
24 }
25
26
27 //=============================================
28 // test  ユーザー定義関数
29 //=============================================
30 void test(void){
31   int hoge=777;
32
33   printf("hoge at test = %d\n", hoge);
34
35 }
```

このプログラムを実行した結果を以下に示す．変数のスコープと優先順位が理解できたなば，この実行結果に納得がいくだろう．

実行結果

```
hoge at main = 999
hoge at test = 777
hoge at for loop = 888
```

2.3 グローバル変数の例

2 つ以上の値を返す関数　これまで学習してきたユーザー定義関数は，返せる値はひとつのみであった．グローバル変数を使うと 2 つ以上の値を返すことができる[2]．その例をリスト 2 に示す．プログラムの大事な点は次のとおりである．

- **6 行目　double value1, value2;**　　こえらの 2 つの変数はグローバル変数で，どこからでも値を読み出したり，書き込んだりできる．
- **32 行目　value1=pow(a,b);**　　a^b を計算する C 言語の関数．

リスト 2: 2 つ以上の値を返す関数をグローバル変数を使って実現する方法.

```
1  #include <stdio.h>
2  #include <math.h>
3
4  void shisuu_kansu(double a, double b);      // プロトタイプ宣言
5
6  double value1, value2;                      // global 変数
7
8  //=================================================================
9  // メイン関数
10 //=================================================================
11 int main(void){
12   double a, b;
13
14   printf("a^bとa^(-b)を計算します\n");
15   printf("a?\t");
16   scanf("%lf",&a);
17   printf("b?\t");
18   scanf("%lf",&b);
19
20   shisuu_kansu(a, b);
21
22   printf("a^b = %f\n", value1);
23   printf("a^(-b) = %f\n", value2);
24
25   return 0;
26 }
27
28 //=================================================================
29 // ユーザー定義関数      a^bとa^(-b)を計算
30 //=================================================================
31 void shisuu_kansu(double a, double b){
32
33   value1=pow(a,b);
34   value2=pow(a,-b);
35
36 }
```

[2]ポインターを使う方が良いが，これは今後の学習範囲

実行結果

```
x^a と x^(-a) を計算します
x?      6.3
a?      1.35
x^a = 11.998024
x^(-a) = 0.083347
```

3 プログラム作成の練習

[練習 1] キーボードから角度 (deg) を入力して，次の 3 つの値を計算して表示させる．

$$\text{value1} = \cos x \qquad \text{value2} = \sin x \qquad \text{value3} = \cos^2 x + \sin^2 x \tag{1}$$

ただし，これらの値はリスト 2 のようにひとつのユーザー定義関数で計算すること．

注意 C 言語の三角関数は，ラジアン単位で計算するので，角度の変換が必要である．

[練習 2] 前回と今回の講義を参考にして，a^x と a^{-x} のグラフを作成する．プログラムは，以下のように動作する．

1. a の値 (`double`:倍精度実数) をキーボードから読み込む．
2. x の範囲をキーボードから読み込む．
3. プロットする点の数を 1000 として，データ間隔 dx を計算する．
4. データ保存用のファイルを開く．
5. ループ文を使って，a^x と a^{-x} を計算して，値をファイルに保存する．a^x と a^{-x} の計算はユーザー定義関数を使うこと．
6. データ保存用のファイルを閉じる．

保存されたデータは次のようになっている．1 列目が x の値，2 列目が a^x の値，3 列目が a^{-x} の値である．ただし，これは $a = 2.3$，範囲 $[-2, 2]$ の場合である．

出来上がったファイル (data.txt) の中身

```
-2.000000 0.189036 5.290000
-1.996000 0.189667 5.272405
-1.992000 0.190300 5.254868
-1.988000 0.190935 5.237390
-1.984000 0.191572 5.219970
-1.980000 0.192211 5.202608
-1.976000 0.192853 5.185304
-1.972000 0.193496 5.168057
-1.968000 0.194142 5.150868
-1.964000 0.194790 5.133735

この辺は長いので，途中省略

1.984000 5.219970 0.191572
1.988000 5.237390 0.190935
1.992000 5.254868 0.190300
1.996000 5.272405 0.189667
2.000000 5.290000 0.189036
```

そして，以下のコマンドを打つことにより，作成したデータをプロットする．

gnuplot によるデータのプロット

```
$ gnuplot
gnuplot> plot "data.txt" using 1:2 with line, "data.txt" using 1:3 with line
```

4　課題

次回の講義の日 (11 月 17 日) の AM8:45 までに，以下の課題をレポートとして提出すること．表紙等は，いつもの通り．表紙のタイトルは「変数のスコープ」とすること．

[問 1] (復予) 教科書の p.162–202 を 3 回読め．重要なところには，赤線あるいは蛍光ペンで印を付けよ．不明な点があれば，クラスの者に聞け．それでも分からない場合は，Office hours を利用して私 (山本) に質問しろ．

[問 2] 本日，配布したプリントを 3 回読め．

[問 3] グローバル変数とローカル変数の違いを，表にまとめよ．

参考文献

[1] 内田智史監修, (株) システム計画研究所編. C 言語によるプログラミング　基礎編　第 2 版. (株) オーム社, 2006.