

式と型

山本昌志*

2006年7月5日

概要

C言語の式の考え方を示す。まずは演算の順序を変える括弧の使い方を学習する。それから複合代入演算子について学ぶ。最後に混合演算や代入時における暗黙の型変換、キャストを用いた明示的な型変換の方法を学習する。

1 本日の授業内容

1.1 前回の復習

先週は教科書の3章の p.73-84 の変数の表示などを学習した。先週の講義で理解すべき内容は、以下の通り。

- 浮動小数¹は、以下のように表すことができる。

```
x=0.00000023456;  
y=2.3456e-7;
```

いずれも同じ値、数学では 2.3456×10^{-7} を表している。この例の e-7 は、 10^{-7} を表している。非常に大きな数や小さな数を表す場合、ゼロを書かなくて済む分、簡単だし間違いが少なくなる。このような表しかたを指数形式と言う。

- 浮動小数の様々な表示方法について、学習した。例えば、`mu=1.25663706144e-6` とする。printf では次のように表示される。

<code>printf("%f\n",mu);</code>	⇒	0.000001
<code>printf("%.10f\n",mu);</code>	⇒	0.0000012566
<code>printf("%20.10f\n",mu);</code>	⇒	0.0000012566
<code>printf("%e\n",mu);</code>	⇒	1.256637e-06
<code>printf("%.10e\n",mu);</code>	⇒	1.2566370614e-06
<code>printf("%20.10e\n",mu);</code>	⇒	1.2566370614e-06

*独立行政法人秋田工業高等専門学校電気工学科

¹整数でない実数と考えてよい。

- %f とすると小数形式で値を表示する。%e とすると指数形式で値を表示する。いずれの場合も小数点以下 6 桁である。
 - %0.nf あるいは%0.ne と表示した場合、それぞれの形式で小数点以下 n 桁で表示する。
 - %m.nf あるいは%m.ne と表示した場合、全体で少なくとも m カラム用意して、小数点以下 n 桁で、それぞれの形式で表示する。カラムと言うのは文字が書ける枠のこと。
- コンピュータプログラムではデータは変数の中に格納される。変数を使うためには、型と変数名を指定—変数の定義—を行わなくてはならない。定義を行うとデータを格納することができるようになる。ただし、変数を定義しただけでは、その中に格納されている値は不定—とんでもない値—である。次のようにすれば、変数の定義と初期化—定義と同時に値の格納—ができる。

```
int hoge=123, fuga=456;
double foo=3.1415, bar=1.6e-19;
```

- 型修飾子 const を付けることにより、定数のように扱うことができる。例えば、次のように変数の定義を行う。

```
const int hoge=123;
const double fuga=9.8765e+43;
```

このように変数を定義すると、プログラム中で変数 hoge や fuga に値の代入ができなくなる。コンパイラーがエラーメッセージを出す。

1.2 本日の学習内容

今回の授業で、教科書 [1] の 3 章「変数と式」の学習を終える。本日の範囲は、p.85-99 である。この範囲のうち、今現在諸君が身につけるべき内容を厳選して、ここでは学習する。

本日の学習のゴールは、以下のとおりである。

- 括弧によって演算の順序を変えることができる—ことを理解する。
- 様々な代入演算子の使い方が分かる。
- 整数型 (int) と倍精度実数型 (double) の両方が含まれた演算では、計算精度の高い方に自動的に変換されることが理解できる。さらに、右辺値は左辺の型に変換されることが理解できる。
- キャストによる明示的な型変換—強制型変換—の使い方が分かる。

2 式を計算するときのテクニック

2.1 括弧

C言語の演算の順序は数学と同じである。積(*)や商(/)は和(+)や差(-)よりも優先度²が高く、先に計算する。まだ学習する必要はないが、C言語にはいろいろな演算子がある。教科書 [1] の表 3.16(p.94) に全ての演算子とその優先度が書かれている。優先度はしばらく学習すると理解できる。数学の優先度と同じであり、また情報科学やコンピューター言語の常識の範囲である。

演算子の優先度を変えたければ、数学同様に括弧を用いることができる。通常、数学では対応が一目で分かるようにするために、3つの括弧—大括弧[] 中括弧{} 小括弧()—を使うことができる。それに対して、C言語では1種類の括弧—小括弧()—しか使えない。ひとつしかなくても、左右の括弧の対応を付けることが可能である。演算の順序を変えるために、大括弧[] 中括弧{} を使ってはならない。

言うまでもないと思うが、左右の括弧の数は等しくなくてはならない。演算子の順序の変更以外にも括弧を使うが、必ず左右で一組である。コンパイラーは厳密に、左右の括弧の対応をチェックする。もし、対応がおかしければコンパイルエラーのメッセージを出し、コンパイルを中止する。

2.1.1 代入演算子

ここは、教科書の p.87-88 あたりに述べていることである。

単純代入演算子 いままで、C言語のイコール(=)—単純代入演算子—と数学のイコールとはまったく異なる働きをするとしつこく述べてきた。

- C言語のイコールは、右辺の式を計算して、左辺の変数に代入する—という操作の命令を表している。したがって、左辺は、必ず変数となる。
- 数学のイコールは、左辺と右辺が等しい—ということを表している。

C言語の単純代入演算子(=)の動作を図1に示す。

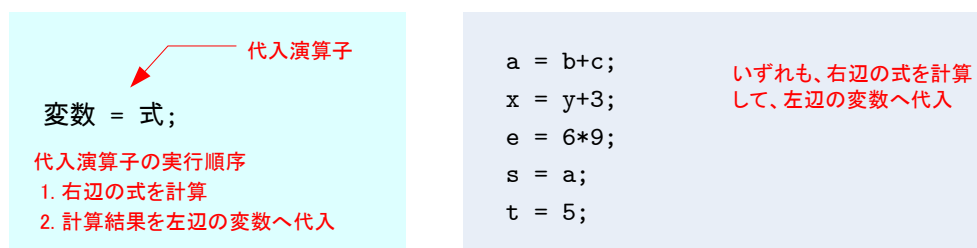


図 1: C言語のイコール—代入演算子—の動作

C言語の左辺の値を左辺値 (lvalue)、右辺の値を (rvalue) と言う。これは憶えておいた方が良い。ときどきコンパイラーのエラーメッセージの中にこれらの単語が使われている。

²積や商は和や差よりも先に計算するので、優先度が高い—と言う。

複合代入演算子 代入演算子は、単純代入演算子と複合代入演算子がある。複合代入演算子とは、教科書の表 3.13 の単純代入演算子 (=) をのぞいた全てである。この中で、現在の諸君が憶えなくてはならないものを表 1 にしめす。

表 1: 憶えるべき複合代入演算子。表中の a の値は、演算の結果である。演算の前の a の値は 7, hoge は 3 とする。いずれも整数型の変数とする。

複合代入演算子	動作	例	単純代入演算子の記述	a の値
+=	加算して代入	a+=hoge	a=a+hoge	10
-=	減算して代入	a-=hoge	a=a-hoge	4
=	積算して代入	a=hoge	a=a*hoge	21
/=	除算して代入	a/=hoge	a=a/hoge	2
%=	剰余算して代入	a%=hoge	a=a%hoge	1

2.2 暗黙の型変換

ここは、教科書の p.95-97 に述べていることである。

混合演算 今までは、同じ型どうしの演算で、結果も同じ型に代入した。しかし、実際の計算では不都合が生じることがある。例えば、整数型と倍精度実数型の加算が必要になることがある。このように、型が異なるものの演算を混合演算と言う。この場合、精度の高い方に型の変換が行われる。整数型の値は倍精度実数型に自動的に変換されて、計算が行われる。常識の範囲でちゃんと計算が行われるのである。

代入における型変換 代入演算において、右辺値と左辺値の型が異なる場合はどうなるか?。この場合は、右辺値は左辺の変数の型に自動的に変換される。これも常識通りで、おかしい値にならないようになっている。

2.3 明示的型変換

ここは、教科書の p.98-99 に述べていることである。

強制的にプログラマーがデータの型の変換を行わなくてはならないことがある。これが必要なほとんどの場合は、整数どうしの除算である。例えば、クラスの英語の点数の合計を格納している整数型の変数 sum の値が 3130 とし、その人数は整数型の変数 n に 41 と格納されているとする。その平均を倍精度実数型の変数 ave に格納したい。

次の例は、間違いである。

```
ave=sum/n;
```

このようにすると，ave の値は 76.000000... となる．右辺は整数と整数の演算なので，右辺値は整数の 76 になる．それを，倍精度実数型の変数 ave に代入するので暗黙の型変換により，76.000000... となるのである．

正しくは，つぎのようにする．

```
ave=(double)sum/n;
```

変数 sum の前に (double) を付けることにより，強制的に倍精度実数型に変換している．中括弧と型名を書くこと，その右の値の型を強制的に変えることができる．これを明示的型変換³言う．この型変換により，sum の値は 3130.000000... の倍精度実数になる．倍精度実数と整数の複合演算なので，整数部分は倍精度実数に変換される．従って，右辺値は 76.341463... となる．これが左辺の ave に代入されるので，ちゃんと平均値が計算できる．

もちろん，

```
ave=sum/(double)n;
```

としても良いし，

```
ave=(double)sum/(double)n;
```

でも問題ない．

しかし，

```
ave=(double)(sum/n);
```

はダメである．これは，括弧内の sum/n が整数どうしの演算なので，この括弧内の値は 76 となる．それを倍精度実数型に変換しているため，右辺値は 76.000000... となる．

3 プログラム作成の練習

[練習 1] 以下の 2 つの式を計算し，結果を表示するプログラムを作成せよ．

$$(\sqrt{2} + \sqrt{3}) \times (\pi + 1.23456) \quad \{[\sin(\pi/6) + \cos(\pi/4) \times \tan(4\pi/3)] + \pi\} / 8$$

[練習 2] 以下の代入演算子の値が等しいことを確認せよ．ただし，a=7，b=3 とする．いずれも整数型の変数である．ひとつのプログラムで，一度に確認するプログラムを書け．

- | | | |
|-----|------|-------|
| (1) | a+=b | a=a+b |
| (2) | a-=b | a=a-b |
| (3) | a*=b | a=a*b |
| (4) | a/=b | a=a/b |
| (5) | a%=b | a=a%b |

³強制型変換とも言う．キャスト (cast: 鑄造する) と言う．

[練習 3] 以下の計算の結果，代入された値を予想せよ．予想の確認を行うためのプログラムを作成し，実行してみよ．

- (1) $\pi+3$ の結果を倍精度実数型の変数に代入する．
- (2) $\pi+3$ の結果を整数型の変数に代入する．
- (3) $9/10$ の結果を整数型の変数に代入する．
- (4) $9/10$ の結果を倍精度実数型の変数に代入する．
- (5) $9*1.0/10$ の結果を整数型の変数に代入する．
- (6) $9*1.0/10$ の結果を倍精度実数型の変数に代入する．

[練習 4] 以下の 10 個の整数の平均値を求めよ．ただし，キャストを使ったプログラムにすること．

123	234	345	456	789
246	468	135	369	555

4 課題

次回の講義の日 (7月12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。

来週からの4章の学習をはじめ。その内容はこれまでに比べかなり難しくなる。予習と復習はかなり重要である。今回のレポートの内容は、来週の授業の準備である。ちゃんと自分の頭で一度考えて、授業に望むと良く理解できるはずである。

- [問1] 教科書の p.102-117 を5回、繰り返し読め。そして、理解できない内容を簡単に箇条書きせよ。ただし、落とし穴 (p.108) や coffee break (p.111) の内容は分からなくても良い。
- [問2] 条件式とその値について説明せよ (p.103)。
- [問3] AND(かつ) と OR(または) と否定について説明せよ (p.107)。
- [問4] 英語の辞書で、if と else の意味を調べ、まとめよ。
- [問5] ブロックと副文について、説明せよ (p.113)。
- [問6] つぎのプログラムの動作を説明せよ。

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int tensu;
6
7     printf("テストの点数?\t");
8     scanf("%d",&tensu);
9
10    if(tensu < 50){
11        printf("いわゆる欠点です。がんばらないと留年します。\\n");
12    }else{
13        printf("合格点です。しかし、油断は禁物です。\\n");
14    }
15
16    return 0;
17 }
```

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.