

第1章 「まずは慣れよう」のまとめ

1.1 用語のまとめ

コンパイル (教 p.2)

とは、ソースファイル（人間が読む／書く形式、C言語の場合拡張子は `.c`）を実行ファイル（CPUが直接理解できる形式、Windows上では拡張子は `.exe`）などに、翻訳することである。

注釈（コメント） (教 p.3旧4新)

とは、ソースファイル中の人間向けのメッセージで、コンパイラは無視する部分である。C言語では「`/*`」から「`*/`」までが注釈である。さらに新しいC言語の仕様(C99)では「`//`」から行末までという形も利用できる。（Visual Studio Expressでも使用可能）

`printf` (教 p.4旧6新)

は、表示を行うための関数である。関数とは定義済みのプログラム部品である。関数呼出しは処理の依頼であり、その時に渡すデータを `引数` という。

文 (教 p.4旧7新)

の末尾には、通常セミコロン「`;`」が必要である。「`{`」と「`}`」の間に置かれた文は上から（同一行に複数文があるときは左から）順次実行される。

`printf`の変換指定のまとめ (教 p.6旧8新) (教 p.354旧378新)

`printf`の第1引数のなかで、「`%`」から始まる部分は変換指定と言い、第2引数以降の値に順に置き換えられる。整数（10進数）を表示するための変換指定は「`%d`」であり、浮動小数点数の表示は「`%f`」を使う。

文字列リテラル (教 p.9旧11新)

とは、一連の文字を二重引用符「`"`」～「`"`」で囲んだものであり、文字の並びを表す。一重引用符「`'`」～「`'`」は別の用途があるため、この目的には使用できない。

拡張表記 (教 p.9旧11新)

文字列中の「`\n`」は `改行`、「`\a`」は警報（ベル）、「`\t`」は `タブ` を表す。このような、「`\`」を使った書き方を拡張表記という。その他の拡張表記については、教科書 p.234 を参照すること。

変数 (教 p.10旧12新)

とは、数値などのデータを収納するための「箱」（メモリの中の場所）である。C言語では、変数を使うためには事前に宣言が必要である。

```
1 int vx;          /* int (整数) 型の変数 vxの宣言 */
2 double fx;      /* double (倍精度浮動小数点数) 型の変数 fxの宣言 */
3 int vx, vy;     /* int (整数) 型の変数 vxと vyの宣言 */
```

Q 1.1.1 C言語の変数にはどのような名前をつけることができるか？（→教 p.102^旧108^新）

代入 (教 p.11^旧13^新)

とは、変数の値を書き換えることである。「変数 = 式」という形式で、右辺の式の値を左辺の変数に代入する。（代入される変数は必ず左辺に書く。「57 = vx」とは書けない。）

初期化子 (教 p.12^旧15^新)

変数の生成（宣言）のときに値を入れることを初期化という。変数は次の形で初期化することができる。（初期化されないときは“不定値”が入る。）

```
型名 変数名1 = 初期化子1, ... , 変数名n = 初期化子n;
```

初期化子 (initializer) は今のところ“式” (expression) と思っておいて良い。（あとで配列を紹介するとき、初期化子と式が異なる場合が出てくる。）

scanf 関数 (教 p.14^旧16^新)

とは、キーボードから数値などデータを読み込むための関数である。

```
1 scanf("_____", &no); /* キーボードから整数を変数 noに読み込む */
```

「&」は「アンパサンド」と読む。詳しくは、ポインタのところの説明する。
(書き方に注意)

puts 関数 (教 p.16^旧18^新)

は、文字列を出力し、最後に改行を行う。printfと異なり、書式変換は行わない。

1.2 文法のまとめ

式 (expression) とは

これまでのところ、

分類	一般形	補足説明
変数		x, i など
整数リテラル		1, 0, 100, 0xff など
文字列リテラル	"~"	"Hello\n" など
関数呼出し	関数名(式, …, 式)	printf("Hello\n") など

宣言 (**declaration**) とは

これまでのところ、

分類	一般形	補足説明
変数宣言	型 変数名 = 初期化子, … , 変数名 = 初期化子;	型は int, double など
