

### 3. 関数 – 引数と戻り値の利用 –

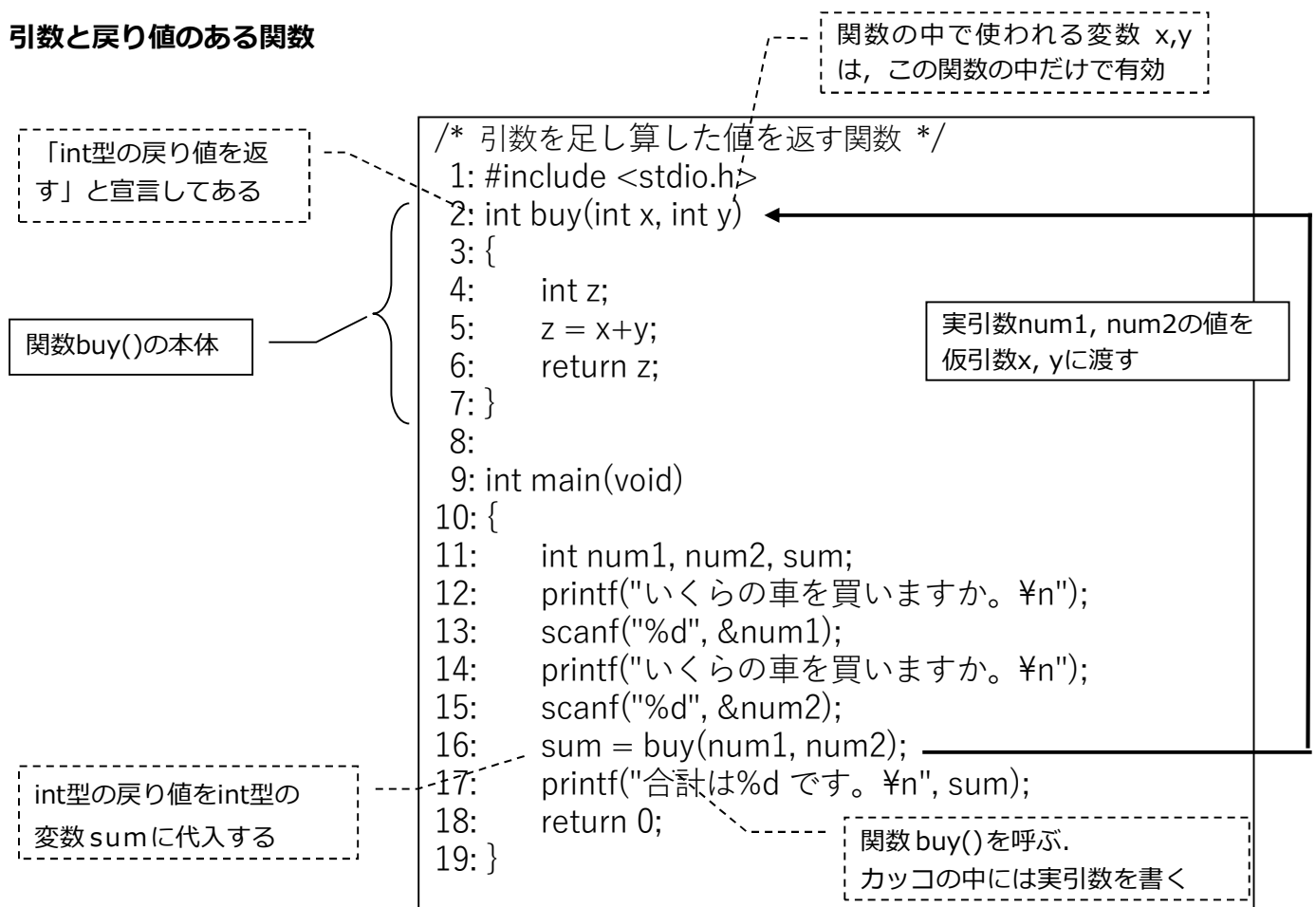
・関数はプログラムの部品であり、それぞれを 1 つにまとめ、大きなアプリケーション（プログラム）が生成される。しかし、単純に関数同士を上下にくっつけても互いの機能を利用することはできない。関数同士を 1 つにまとめ、それぞれの関数の機能を利用して、より大きなプログラムにするには関数間でデータのやり取りが必要になる。このやり取りを担っているのが**引数**（実引数と仮引数）と **return** 文による**戻り値**である。

・戻り値は関数定義部（関数本体）から関数呼出し元へ流れるデータのことである。引数とは異なり、呼出し元に返すデータは 1 つだけである。

・戻り値を利用する場合、関数定義部では **return** 文で指定したデータの型を関数名の左隣に宣言する。関数呼出し元では戻り値と同じ型の変数を用意しておき、代入文により戻り値を受け取る方法が一般的である。

・戻り値を利用しない場合、関数定義部では関数名の左隣に「void」と宣言する。関数呼出し元では戻り値を利用するような文を記述してはいけない。

#### ・ 引数と戻り値のある関数



3.1 Moodleのfp6-3-1.cppをダウンロードし、以下のようなプログラムに変更しなさい。

「関数で求めた結果を関数の呼出し元（この場合はmain()関数）に返却し、関数の呼出し元でその結果を表示する」

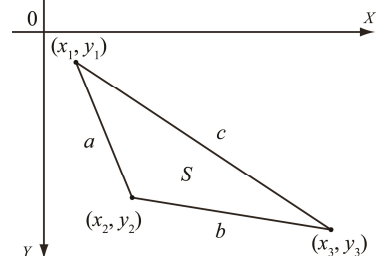
3.2 整数  $x$  に対し、 $x^2 + 2x + 5$  の式を計算し、その計算値を返す関数 `func1()` を作成せよ。この関数を用いて、キーボードから入力された  $x$  の値に対する計算値を出力するプログラムを作成せよ。

3.3 2.1の関数を標準体重が戻り値になるように変更し、標準体重を求め表示するプログラムを作成せよ。

3.4 実数値 $x$ を入力し、べき乗を求めるプログラムを作成せよ。ただしべき乗の計算 ( $x^n$ :  $n$ は整数) は自身で定義した関数を再帰的に呼び出すことで実現すること。

3.5 3つの座標  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  を入力し、それが三角形の三辺になるかどうか調べよ。さらに三角形になるのであれば、その面積  $S$  を求めるプログラムを作成せよ。ただし、三角形の面積はヘロンの公式を使用して求めること。

・座標を受け取り、2点間の距離を返す関数、  
・面積を求めるための関数、  
をそれぞれ定義し、プログラムを作成すること。



・三角形が成立する条件:  $a + b > c > |a - b|$

・ヘロンの公式

$a, b, c$ : 三角形の各辺の長さ

$$S = \sqrt{s(s-a)(s-b)(s-c)}, \text{ ただし, } s = \frac{(a+b+c)}{2}$$

```
三角形の頂点の座標を入力
頂点A(x1, y1) : 1 1
頂点B(x2, y2) : 1 6
頂点C(x3, y3) : 7 1
三角形の面積は 15.00 です
```

### 第3章「関数－引数と戻り値の利用－」のチェック項目

#### 戻り値について

- ☐ return文を利用して戻り値を指定できた
- ☐ 関数名の左隣に記述する戻り値の型を正しく指定できた
- ☐ 引数とは異なり、戻り値は1つだけであることを理解した
- ☐ 戻り値は定数や変数、式などが指定できることを理解した
- ☐ return文が実行されると処理が関数呼び出し元に戻ることを理解した
- ☐ 関数の呼び出し元で戻り値が利用できることを理解した
- ☐ 呼び出し元で戻り値を利用するには,
  1. 呼び出し元で宣言した戻り値と同じ型の変数に代入する,
  2. 関数呼び出しの記述を直接利用する,いずれかの方法で戻り値を利用することができた