

# プログラミング基礎 #04

## 条件分岐

担当：向井 智彦

# 前回のまとめ

- 変数：1つのデータを一時保存できる
  - 変数宣言 「int a;」, 代入(上書き) 「a = 10;」
  - 変数からの代入(読み出し) 「int b = a;」
- 配列：同じ型の複数の変数を添字付きで格納
  - 配列宣言 「int a[10];」
  - 配列の先頭番号は 0 「a[0] = 1;」...「a[9] = 10;」
- 変数を用いた演算
  - 「a = a + b + 10 \* a;」など代入演算子 = の左右に同じ変数が登場する際の動作

# おさらいクイズ

以下のプログラムの出力は以下の解答群のうち、どれ？

A: 0

B: 0.2

C: 1

D: 2

```
#include <iostream>
int main()
{
    int a = 0;
    a = a + 10;
    double b = a * 5;
    double ab = a / b;
    std::cout << ab << std::endl;
}
```

# 本日の内容

- 目標：簡単な数字当てゲームの作成
- 講義内容：条件による分岐処理
  - ブール型変数 `bool`, `true`, `false`
  - 条件分岐 `if () { ... } else { ... }`
  - 標準入力ストリーム `cin`
  - プログラムファイルとコンパイル(?)

# @ wandbox.org

```
#include <iostream>
using namespace std;
int main() {
    int x = 5, y = 6;
    if (x > y) {
        cout << x << " is larger than " << y << endl;
    }
    else {
        cout << x << " is smaller than " << y << endl;
    }
}
```

# 文法図解

```
#include <iostream>
using namespace std;
int main() {
    int 条件式 y = 6;
    if (x > y) { 条件式が成立するときに実行される文
        cout << x << " is larger than " << y << endl;
    }
    else { 条件式が成立しないときに実行される文
        cout << x << " is smaller than " << y << endl;
    }
}
```

# ブール(**bool**)型

- 真理値 とか 真偽値, ブーリアン型とも
  - 真: **true** あるいは偽: **false**
  - true の否定 は false, falseの否定はtrue
    - 数学「命題・論理」と関係(学んでなくてもOK)

```
bool a, b;  
a = true;  
b = !a; // 否定演算  
c = a * 2 + b * 2; // あり得ない演算  
cout << a << ", " << b << ", " << c << endl;
```

# 比較演算：演算結果はbool型

※以下,  $a$  と  $b$  は任意の型 (int, char, double, bool, ...)

- 一致「 $a == b$ 」(代入「 $=$ 」と混同しないよう注意)
  - $a$  と  $b$  が同じ値であれば true, 違うなら false
- 非一致「 $a != b$ 」
  - $a$  と  $b$  が異なる値であれば true, 違うなら false
- 大なり「 $a > b$ 」, 以上「 $a >= b$ 」
  - $a$  が  $b$  より大きい値なら true, 小さいなら false
- 未満「 $a < b$ 」, 以下「 $a <= b$ 」
  - $a$  が  $b$  より小さい値なら true, 大きいなら false



# 比較演算の具体例

```
bool quiz[5];  
quiz[0] = 10 != 2;  
quiz[1] = 3.0 == 3.0;  
quiz[2] = 10 < 2;  
quiz[3] = [5.0 * 2.0] >= 20;  
quiz[4] = !(8 + 2.0 > [10 * 2]);
```

[比較演算より早く実行される算術演算]

# 真偽値の論理演算

※ a と b はどちらもbool型

- 否定「 !a 」 「 !b 」
  - true の否定は false, false の否定は true
- 論理和「 a || b 」
  - a と b のいずれか true なら, 結果は true
  - a と b の両方が false の時, 結果は false
- 論理積「 a && b 」
  - a と b の両方が true なら, 結果は true
  - a と b のどちらかが false なら, 結果は false

# 論理演算の具体例

```
bool t = true, f = false;  
bool quiz[4];  
quiz[0] = t && f;  
quiz[1] = t || f;  
quiz[2] = f && (t || t);  
quiz[3] = (f && f) || t;
```

# 比較演算と論理演算

```
int a = 1, b = 2, c = 3;  
bool quiz[4];  
quiz[0] = a < b && b <= c;  
quiz[1] = a <= b || a < c;  
quiz[2] = [a - b > 0] && [c - a > 0];  
quiz[3] = [a > 0] && [b > 0] && [c > 0];
```

[論理演算より早く実行される比較演算]

# 条件分岐 if & else

- 真偽値true/falseに応じて異なる処理を行う、処理分岐のための制御構文

```
int main() {  
    int 条件式: 真偽値: 比較演算 & 論理演算  
    if (x > y) { 条件式がtrueの時に実行される文  
        cout << x << " is larger than " << y << endl;  
    }  
    else { 条件式がfalseのときに実行される文  
        cout << x << " is smaller than " << y << endl;  
    }  
}
```

# 条件分岐の具体例

```
double x = 11.0, y = 13.0;
if (x == 10.0 && y < 20.0) {
    cout << "A" << endl;
} else if (!(x >= 10.0) || y >= 10.0) {
    cout << "B" << endl;
} else
    cout << "C" << endl;
}
```

青条件式が満たされず, 緑条件式が成り立つなら

# まとめ

- bool型
  - true と false のいずれかのみ
- 比較演算
  - 数値の等しさ・大小関係をbool型で評価
- 論理演算
  - bool型変数に対する論理和・論理積
- 条件分岐
  - bool値に応じて異なる処理を行う仕組み

# 講義しなかったこと

- 論理演算「&」「|」と「&&」「||」との違い
- 排他的論理演算「^」
- switch 文
  - 整数値に応じた条件分岐
    - (if文はブール値に応じた分岐)



# 演習課題01：球体の諸特徴量

提出期限：10/29(月)、ファイル名：04-01.cpp

- 入力された球の半径に対し、球の表面積より体積が大きい場合は「Volume is larger.」、体積より表面積が大きい場合は「Area is larger.」、同じ場合は「Equal!」と出力するプログラムを作成
  - 球の半径をcinより取得
  - 球の表面積と体積を計算
  - 両者の比較し結果に応じたメッセージを出力
  - ヒント：「Equal!」が決して出力されないこともある
    - コンピュータが扱う浮動小数には誤差があるため

# 演習課題02：ソート

提出期限：10/29(月)、ファイル名：04-02.cpp

- 入力された3つの浮動小数を大きい順に並べ替えて出力するプログラムを作成
  - 3つの整数を半角スペース区切りで入力
  - 例：「4 -1 200」
  - ヒント：バブルソート
    - 隣り合う2つの数字が「 $<$ 」なら入れ替え (hint. 03-01)
    - 最低3回は大小関係进行比较する
- 4つ以上の浮動小数も扱えるように拡張したら加点

# 演習課題Extra(やらなくてもOK)

提出期限: 10/29(月)、ファイル名: 04-ex.cpp

- プログラム中に隠された正の整数を推理するゲームを作成
  - プレイヤーの入力はcinより取得
  - 入力した数字が正解より大きい小さいか、ヒントメッセージとして出力
    - メッセージは任意
  - ヒントメッセージの工夫に対して加点
    - 例: 数字そのものの大小関係に加え、約数の数の大小関係も出力する、たまにウソをつく、などなど
    - 工夫した点をプログラムコード中に **// コメント**

# 標準入力ストリーム cin

```
# include <iostream>
int main()
{
    int x, y;
    // キーボード入力を変数x, yの順に代入
    std::cin >> x >> y;
    std::cout << x << ", " << y;
}
```

```
12 # include <iostream>
13 int main()
14 {
15     int x, y;
16     // キーボード入力を変数xに代入
17     std::cin >> x >> y;
18     std::cout << x << " " << y;
19 }
20
```

```
$ g++ prog.cc -Wall -Wextra -I opt/wandbox/boost-1.68.0/gcc-head/include -std=gnu++2a
```

Stdin

265 334

wandboxでは「Stdin」の欄に記入  
複数の数字は空白スペースを挟んで記入

Run (or Ctrl+Enter)

#5

Share

Code

#4

Start

265 334

# 来週の予告

- 反復 (for, while)
- 変数のスコープ
- プログラムファイルとコンパイル