

## 問題1 (提出期限 本日 15:00) Struct の定義, Struct データの入出力と計算

以下は, 二人の学生の試験の成績を比較したり, 平均点を出したりするプログラムである. 学生のデータは, 学籍番号 (整数), 氏名 (文字列), 試験の点数(100 点満点の整数)から構成される. (1) このデータをまとめた Struct Student を定義せよ. (2) 実行例のように, 二人の学生のデータをキーボードから入力し, (3) 高得点の学生のデータを表示せよ. ただし, 同点の場合には, 2 名とも表示する. (4) 最後に, 二人の学生の平均点を表示する.

プログラム実行例に合うように空欄(1)-(4)を適切に埋めてプログラムを完成させよ. (プログラム全体を ex9-1.cpp として提出すること)

```
// ex9-1.cpp
#include <iostream>

(1) struct Student を定義せよ.

int main() {
    Student s1,s2; // 二人の学生の変数

    (2) s1, s2 のデータをキーボードから入力

    (3) 高い得点の学生のデータを表示, 同点の場合は, 両方を表示

    (4)平均点を表示

    return 0;
}
```

実行例 (下線灰色はキーボード入力の跡, プログラムからの出力ではないことに注意!)

```
% ./ex9-1
Input num, name and score for student1 ! 212999 SeikeiTaro 67
Input num, name and score for student2 ! 212888 MomonoMichi 82
Top Score: 212888 MomonoMichi 82
Average score: 74.5

% ./ex9-1
Input num, name and score for student1 ! 212777 SakuraMomoko 75
Input num, name and score for student2 ! 212678 KeyakiJiro 75
Top Score: 212777 SakuraMomoko 75
Top Score: 212678 KeyakiJiro 75
Average score: 75
```

## 問題 2 (提出期限 12 月 5 日 23:59) struct 利用関数, cin と cout 対応

以下のプログラムは、二次元座標  $x, y$  (実数) をメンバーに持つ struct Point を利用し、二次元座標点の入出力, 2 点間の中点の計算, 3 点で囲まれる三角形の面積の計算などを行うプログラムである。空欄 (1)-(6) をプログラムの実行例に合うように適切に埋めてプログラムを完成させよ。(プログラム全体を ex9-2.cpp として提出する) 注意: 既存のコードに加えることはできるが, 変更や削除をしてはならない。

```
#include <iostream>
# include <cmath>    // cmath にある 実数の絶対値を返す fabs ( ) 関数 を利用する
// (1) 二次元座標  $x, y$  (実数) をメンバーに持つ struct Point を定義せよ。 注意! 教科書とは異なる

// (2) cin から直接 Struct Point へ入力する関数を完成させよ。 教科書 p.105 を参考にして下線を埋める
std::istream& operator>>(std::istream &in, _____ )
{
    _____ }
// (3) cout へ直接 Struct Point を出力する関数を完成させよ。 教科書 p.105 を参考にして下線を埋める
std::ostream& operator<<(std::ostream &out, _____ )
{
    _____ }
// (4) 2 つの Point を引数として受け取り, 中点の Point を計算して返す関数 mid

// (5) 3 つの Point を引数として受け取り, 3 点で囲まれる三角形の面積を返す関数 triangle
3 点で囲まれる三角形の面積 参考 https://keisan.casio.jp/exec/system/1173765469
実数の絶対値を求める関数, std::fabs() 参考 https://cpprefjp.github.io/reference/cmath/fabs.html

int main() {
    // (6) struct Point p1 の宣言, 座標は,  $x=1.2$   $y=3.4$  に初期化せよ。 教科書 p.99
    Point p2, p3, pm; //これらの点はデフォルトで原点になる

    // p3 にキーボードから座標値を入力
    std::cout<<"Input x,y for point p3! ";
    std::cin >> p3;                // 関数(2)を利用

    // p1, p2, p3 の座標値を表示
    std::cout<<"p1: "<<p1<<" p2: "<<p2<<" p3: "<<p3<<"\n"; // 関数(3)を利用

    // p1 と p3 の中点 pm の計算
    pm = mid(p1, p3);              // 関数 mid を利用
    // pm の座標値を表示
    std::cout<<"Middle point pm : "<<pm<<"\n"; // 関数(3)を利用

    // p1 と p2 と p3 で囲まれる三角形の面積を計算
    std::cout<<"Triangle area (p1, p2, p3) is "<<triangle(p1, p2, p3)<<"\n"; //関数 triangle を利用

    return 0;
}
```

実行例 (下線灰色はキーボード入力の跡, プログラムからの出力ではないことに注意!)

```
% ./ex9-2
Input x,y for point p3! 3.4
p1: (1.2, 3.4) p2: (0, 0) p3: (3, 4)
Middle point pm : (2.1, 3.7)
Triangle area (p1, p2, p3) is 2.7
```

```
% ./ex9-2
Input x,y for point p3! -3.4 1.8
p1: (1.2, 3.4) p2: (0, 0) p3: (-3.4, 1.8)
Middle point pm : (-1.1, 2.6)
Triangle area (p1, p2, p3) is 6.86
```

### 問題 3 (提出期限 12月5日 23:59) 構造体 struct の配列 vector, (ファイル利用の復習)

ここで作成するプログラムでは、会員の氏名、体重(kg)、身長(cm)が書かれたファイル(members.txt)からデータを読み、BMI 値を計算し、肥満の度合いを、肥満 (FAT)、痩せすぎ (LEAN)、標準 (STANDARD) の 3 種類に分類してデータベースを作成する。そのあと、指定されたカテゴリの会員のみをデータベースから選んで表示するプログラムである。

次頁の ex9-3\_skel.cpp に示すような一人分の会員データを持つ struct Member を定義し、struct Member の配列 members に、ファイルから会員データ (名前、体重、身長) を読み込む。その際に、BMI 値を計算し、以下の判定式により、判定の string (FAT, LEAN, STANDARD) も設定して、members 配列に挿入して、会員データベースを作成する。

BMI の計算は、身長  $h$  (m 単位)、体重  $w$  (kg 単位) とすると、 $bmi = w/(h*h)$

判定基準	BMI 18.5 未満	LEAN (痩せすぎ)
	BMI 18.5 以上 25 未満	STANDARD (標準)
	BMI 25 以上	FAT (太り過ぎ)

身長  $h$  を cm から  
m に換算すること

次に、キーボードから、カテゴリ (FAT, LEAN, STANDARD) か、全員表示 (ALL)、あるいはプログラム終了(QUIT)のいずれかをを入力し、該当する会員のデータを表示する。QUIT が入力されるまで、繰り返し指定された会員の表示を繰り返す。QUIT が入力された場合には、プログラムを終了する。配布された ex9-3\_skel.cpp ファイルの空欄(1)-(10)を埋めて、ex9-3.cpp プログラムを完成させよ。

実行例 1 **LEAN FAT STANDARD ALL QUIT** それ以外の単語を入力し、確認せよ

```
% ./ex9-3
Input file name = members.txt
Specify category to display (FAT,STANDARD,LEAN, ALL) or QUIT! ALL
----- ALL Members list
      Name   Weight   Height      BMI   Hantei
      yuki    50.3     170.6   17.2826   LEAN
     haruto    87.4     182.3   26.2989   FAT
      souta    65.2     176.2   21.0008 STANDARD
      yuto     56.4     168.2   19.9355 STANDARD
     haruki    83.2     165.3   30.4493   FAT
      koki     75.2     173.5   24.9815 STANDARD
     kaito     63.8     183.2   19.0095 STANDARD
     takumi    72.6     172.3   24.4549 STANDARD
     hayato    53.6     179.5   16.6355   LEAN
     syota     65.7     163.2   24.6675 STANDARD
      taro     56.8     166.5   20.489   STANDARD
     ichiro    94.2     168.1   33.3362   FAT
     jaiko     38.5     145.2   18.2611   LEAN
Specify category to display (FAT,STANDARD,LEAN, ALL) or QUIT! LEAN
----- LEAN Members list
      Name   Weight   Height      BMI   Hantei
      yuki    50.3     170.6   17.2826   LEAN
     hayato    53.6     179.5   16.6355   LEAN
     jaiko     38.5     145.2   18.2611   LEAN
Specify category to display (FAT,STANDARD,LEAN, ALL) or QUIT! Debu
category input error! input again!
Specify category to display (FAT,STANDARD,LEAN, ALL) or QUIT! FAT
----- FAT Members list
      Name   Weight   Height      BMI   Hantei
     haruto    87.4     182.3   26.2989   FAT
     haruki    83.2     165.3   30.4493   FAT
     ichiro    94.2     168.1   33.3362   FAT
Specify category to display (FAT,STANDARD,LEAN, ALL) or QUIT! Quit
category input error! input again!
Specify category to display (FAT,STANDARD,LEAN, ALL) or QUIT! QUIT
%
```

実行例 2 (members.dat がない時)

この実行例は確認しなくてよい

```
% ./ex9-3
cannot open file
```

配布された ex9-3\_skel.cpp ファイルを, cp ex9-3\_skel.cpp ex9-3.cpp でコピーして作成する

```
// ex9-3_skel.cpp

#include<iostream>
#include<iomanip>
#include<fstream>
#include<vector>
using std::cin, std::cout, std::cerr, std::setw;

struct Member // 会員データの struct 定義
{
    std::string name; // 氏名
    double weight; // 体重
    double height; // 身長
    double bmi{0}; // BMI 値 デフォルト値は 0 だが, 体重と身長がわかったら, 計算して設定する
    std::string hantei; // BMI 値により, 判定 "LEAN", "STANDARD", "FAT"のいずれかが入る
};

void print_title() // 表示の際の先頭行の各項目のタイトル表示 全て幅 10 文字で表示
{
    cout<<setw(10)<<"Name"<<setw(10)<<"Weight"<<setw(10)<<"Height"
        <<setw(10)<<"BMI"<<setw(10)<<"Hantei" <<"\n";
}

void print(Member m) //引数の一人分の会員 m のデータを表示
{
    // (10) 引数 m の氏名, 体重, 身長, BMI, 判定を全て幅 10 文字で表示. print_title()を参考にせよ
    (10)
}

int main()
{
    // 1. 会員データファイル members.txt をリードで open,
    std::ifstream fin("members.txt");
    // (1) fin ファイルのエラーチェックを行い, エラーの場合にはプログラム終了. main 関数は 1 を返す
    (1)
    //ファイル名の表示
    cout<<"Input file name = members.txt\n";

    // 2. ファイルから, 名前, 体重, 身長を読み, bmi を計算し hantei も設定, members 配列に登録
    std::vector<Member> members; // 会員データの配列
    Member m; //一人分の会員データ m

    while( (2) ) //(2) fin ファイルから一人分の名前, 体重, 身長を m に読む
    {
        (3) // (3) BMI を計算し, m の bmi に設定
        (4) // (4) BMI 値から, m の hantei 判定に設定

        (5) // (5)会員データ m を会員データ配列 members に加える
    }
}
```

// 3. カテゴリー入力と表示の繰り返し

std::string category; //表示したい会員のカテゴリもしくは、プログラム中止のための Quit

while (true){ //キーボードから終了(QUIT)が入力されるまで、永久ループ

//表示したいカテゴリ、もしくは、プログラム終了(QUIT) を入力

cout<<"Specify category to display (FAT,STANDARD,LEAN, ALL) or QUIT! ";

cin >> category; // キーボード入力

if( category == "QUIT" ) break; // QUIT の時は break で while 終了.

// または、break の代わりに、return 0; で、main 終了でもいい

else if ( (6) ){ // (6) 入力が ALL の時 全員の表示

cout<<"----- " << category <<" Members list¥n"; //表示するカテゴリの表示

print\_title(); // 各項目のタイトル表示

// (7) members の全員のデータを表示 print 関数利用

(7)

}

else if( (8) ) // (8) 入力が表示肥満度指定の時

{

cout<<"----- " << category <<" Members list¥n"; //表示するカテゴリの表示

print\_title(); // 各項目のタイトル表示

// (9) members の該当カテゴリの会員データを表示 print 関数利用

(9)

}

else //入力が指定した単語以外の時は、再度入力に戻る

cout<<"category input error! input again!¥n";

} // while の終了

return 0;

}