

Fundamental Computer Programming- C++ Lab(I)

LAB 2

BMI Evaluation

Week 2, Fall 2020

International Bachelor Program in Informatics

College of Informatics

Yuan Ze University

Purposes

- Get familiar with the basic structure of a C++ program
- Get familiar with two of the three types of control statements
 - Sequence statements
 - Selection statements
 - Repetition statements (covered in next lab)
- Make a faster implementation of the same algorithm
- Develop problem solving skills

Sequence Statements

S1;

S2;

S3;

...

Sn;

Selection Statements (1)

Single Selection **if** statement

```
if (condition) {  
    S1;  
    S2;  
    ...  
    Sk;  
}
```

A **block is defined**. A set of statements contained within a pair of braces, i.e., { }, is called a block.

If there is only one statement in a block, the pair of braces can be omitted.

Double selection **if** statement

```
if (condition) {  
    S1;  
    S2;  
    ...  
    Sk;  
}  
else {  
    ....  
}
```

Selection Statements (2)

- In a multiple selection **if** statement, there is at least one **else if**.

```
if (condition) {  
    S1;  
    S2;  
    ...  
    Sk;  
}  
else if (...) {  
    ...  
}  
else if (...) {  
    ...  
}  
else {  
    ...  
}
```

```
if (sGrade >= 80) {  
    cout << "A";  
    countA = countA + 1;  
}  
else if (sGrade >= 70) {  
    cout << "B";  
    countB = countB + 1;  
}  
else if (sGrade >= 60) {  
    cout << "C";  
    countC = countC + 1;  
}  
else {  
    cout << "F";  
    countF = countF + 1;  
}
```

Example

```
#include <iostream>
using namespace std;
```

```
int main( )
{
    int number1;
    int number2;
    int number3;
    int result1;
    int result2;
    int result3;
    int result4;
    int sum;
    bool resultAvailable = true;
```

```
    cin >> number1 >> number2 >> number3;
    if(number2 == 0 || number3 == 0)
        resultAvailable = false;
    else {
        result2 = number1 / number2;
        result3 = number2 % number3;
        result1 = number1 * number2;
        result4 = number1 + number2 + number3;
        sum = result1 + result2 + result3 + result4;
    }
    if(resultAvailable)
        cout << sum << endl;
    else cout << "Results are not available!" <<endl;

    return 0;
}
```

Problem: Body Mass Index Calculator

- Inputs: Weight of a person in kilogram and Height in meter (real numbers).
- Calculate the person's BMI (Body Mass Index).
- Outputs: The person's weight, height, BMI value, a BMI level statement, etc. BMI should also be a real number.

Basic: BMI Evaluation (70%)

- Body Mass Index Calculator

$$BMI = Weight/Height^2$$

- Determine a BMI level statement

BMI level	BMI range(kg/m ²)
Unreasonably small BMI	BMI <=10
Highly severely underweight	10 < BMI <= 15
Severely underweight	15 < BMI <= 16
Underweight	16 < BMI <= 18.5
Normal	18.5 < BMI <= 25
Overweight	25 < BMI <= 40
Obese	40 < BMI <= 50
Severely obese	50 < BMI <= 60
Highly severely obese	60 < BMI <= 70
Unreasonably large BMI	BMI > 70

Input and Output Format

■ Input Format

Input weight (kg): 65.2
Input height (m): 1.73

Typed in from
keyboard

Printed out on the screen by your program.
They are called **prompting messages**.

■ Output Format

Your weight(kg): 65.2

Your height(m): 1.73

Your BMI: 21.7849

BMI value

Your BMI level is normal

pound sign space

BMI level

More Examples

```
Input weight (kg): 18.6  
Input height (m): 2.3  
Invalid weight ( $\geq 20\text{kg}$  and  $\leq 200\text{kg}$ )
```

```
Input weight (kg): 67.8  
Input height (m): 2.9  
Invalid height ( $\geq 0.9\text{m}$  and  $\leq 2.5\text{m}$ )
```

```
Input weight (kg): 89  
Input height (m): 1.63  
# Your weight(kg): 89  
# Your height(m): 1.63  
# Your BMI: 33.4977  
# Your BMI level is overweight
```

```
Input weight (kg): 43.8  
Input height (m): 1.87  
# Your weight(kg): 43.8  
# Your height(m): 1.87  
# Your BMI: 12.5254  
# Your BMI level is highly severely underweight
```

Other Materials Needed

- You have to define the variables used in the program as “double” for storing real numbers. For example:
double myWeight;
double myHeight;
double myBMI;
- You may use some Boolean (logical) operators to express a condition in if (...) statement. For example:
if(myWeight >= 30.3 && myHeight <= 100) {
...}
else if (myWeight <= 50 || myHeight != 200.8) {
...}
else {
...}
- **&&** is logical AND. X && Y is true only when X and Y are both true.
- **||** is logical OR. X || Y is true only when X or Y is true.
- Add “include <cmath>” into the program because you may use a function called sqrt to calculate the square root of a number.

Constraints on Input Data

- Weight should be greater than or equal to 20kg and less than or equal to 200kg.
- Height should be greater than or equal to 0.9m and less than or equal to 2.5m.
- If weight or height is not in the defined range, a **prompting message** should be printed.
 - “Invalid weight ($\geq 20\text{kg}$ and $\leq 200\text{kg}$) “ should be printed for an **invalid weight**.
 - “Invalid height ($\geq 0.9\text{m}$ and $\leq 2.5\text{m}$)“ should be printed for an **invalid height**.
 - **After printing the message , your program should stop execution.**

Follow All Requirements

- Input formats
- Output formats
- Must use multiple selection if (...) else if (...) to a great extent, at least eight times of **else if**. If not, the score is reduced by 10 points.
- Coding styles
 - Avoiding using variables which do not have expressive power. That is, a variable name should carry the meaning of an object in which the variable intends to represent.

If you don't follow the requirements, up to 30% of the points for your lab will be deduced.

Further: Give Advices (30%)

- If a person's BMI is not normal and assume his/her height is fixed, also print out an advice about his/her weight which will bring the person's BMI to 25 if the person's BMI is larger than 25 or bring the value to 18.5 if the person's BMI is smaller than 18.5. An output example should look as follows.

Your weight(kg): 150

Your height(m): 1.7

Your BMI: **51.9031**

Your BMI level is **Highly severely obese**

Decrease your weight to xxx.xx kg to bring your BMI value to 25

Note: The output of the last line in this example should be “**Increase your weight to xxx.xx to bring your BMI value to 18.5.**” if a person is underweight. You should also place a # in the beginning of an output line.

An Example of This Case

```
Input weight (kg): 33
Input height (m): 1.7
# Your weight(kg): 33
# Your height(m): 1.7
# Your BMI: 11.4187
# Your BMI level is highly severely underweight
# Increase your weight to 53.465 kg to bring your BMI value to 18.5
```

```
Input weight (kg): 160.4
Input height (m): 1.9
# Your weight(kg): 160.4
# Your height(m): 1.9
# Your BMI: 44.4321
# Your BMI level is obese
# Decrease your weight to 90.25 kg to bring your BMI value to 25
```

Bonus: Making Sure Reasonable Input Data (+20 points)

- It is possible that weight and height are both in defined ranges. But a calculated BMI value might be unreasonably small or large due to wrongly given weight or height. In this situation, your program should print a message “**Weight or height might be wrongly given.**” Then, we can use the following rule to give a suggestion.
 - Assume (假設) the given weight is reasonable, and the given height is unreasonable, then print out “If given weight X kg is correct, then the given height should be smaller (or greater) than Y.” Where X is the value of the given weight and Y is a height value. You can have either “smaller” or “greater” in the printout.
 - Assume the given height is reasonable, and the given weight is unreasonable, then print out “If given height X m is correct, then the given weight should be smaller (or greater) than Y kg.” Where X is the value of given height and Y is a weight value. You can have either “smaller” or “greater” in the printout.

Examples of This Case

```
Input weight (kg): 34.8
Input height (m): 2.1
# Your weight(kg): 34.8
# Your height(m): 2.1
# Your BMI: 7.89116
Weight or height might be wrongly given.
If given weight 34.8 kg is correct, then the given height should be smaller than 1.86548 m.
If given height 2.1 m is correct, then the given weight should be greater than 44.1 kg.
```

```
Input weight (kg): 154
Input height (m): 1.2
# Your weight(kg): 154
# Your height(m): 1.2
# Your BMI: 106.944
Weight or height might be wrongly given.
If given weight 154 kg is correct, then the given height should be greater than 1.48324 m.
If given height 1.2 m is correct, then the given weight should be smaller than 100.8 kg.
```

Rules for Program Submission

- Put all the relevant files in the same folder.
- Name your folder SID_LabX, where ID is your student ID number and X is the number assigned to the lab. If a lab has N parts, $N > 1$, then create N sub-folders with their names SID_LabX_N in the the folder SID_LabX.
 - For example, for Lab 2 with only one part and with student ID number 1041544, the name of the folder must be S1041544_Lab2. N is omitted if there is only one part.
 - Another example, similar to the above but Lab 2 has two parts. Then, you have to create a folder S1041544_Lab2 and two sub-folders S1041544_Lab2_1 and S1041544_Lab2_2
- Compress the folder into a file named SID_LabX.zip, for example, S1041533_Lab2.zip. Then, submit the compressed file
- If you violate this rule, your lab will not be graded. If graded other penalty will be applied.