Programming with C++

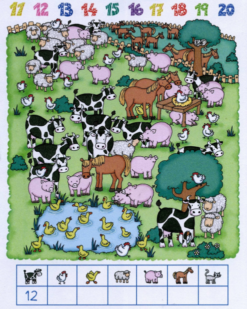
COMP2011: Examples on C++ Basics and Controls

Cecia Chan
Cindy Li

Department of Computer Science & Engineering
The Hong Kong University of Science and Technology
Hong Kong SAR, China

Part I

Count Animals



The Count Animals Problem

- There are two types of animals, pigs and sheeps in a farm.
- Each pig weighs 4.5 units and each sheep weighs 3 units.
- The total weight of animals in a barn should be exactly 36 units.
- List out all possible combinations of bigs and sheeps in the farm.

Solution:

$$\begin{aligned}
0 * 4.5 + 12 * 3 &= 36 \\
2 * 4.5 + \ \ 9 * 3 &= 36 \\
4 * 4.5 + \ \ 6 * 3 &= 36 \\
6 * 4.5 + \ \ 3 * 3 &= 36 \\
8 * 4.5 + \ \ 0 * 3 &= 36
\end{aligned}$$

First Attempt: What's Wrong?

```cpp
#include <iostream>      /* File: two-animals-v1.cpp */
using namespace std;

int main()
{
    float wt_pig = 4.5;
    float wt_sheep = 3.0;
    float total_wt = 36;

    for (int num_pigs = 0; num_pigs * wt_pig <= total_wt; num_pigs++)
    {
        float remain_wt = total_wt - num_pigs * wt_pig;
        int num_sheeps = remain_wt / wt_sheep;
        remain_wt = remain_wt % wt_sheep;

        if (remain_wt == 0)
            cout << num_pigs << " * " << wt_pig << " + "
                 << num_sheeps << " * " << wt_sheep << " = "
                 << total_wt << endl;
    }
    return 0;
}
```

Second Attempt: Any Problem?

```
#include <iostream>     /* File: two-animals-v2.cpp */
using namespace std;

int main()
{
    float wt_pig = 4.5;
    float wt_sheep = 3.0;
    float total_wt = 36;

    for (int num_pigs = 0; num_pigs * wt_pig <= total_wt;
    ↪   num_pigs++)
    {
        float remain_wt = total_wt - num_pigs * wt_pig;
        int num_sheeps = remain_wt / wt_sheep;
        remain_wt -= num_sheeps * wt_sheep;

        if (remain_wt == 0)
            cout << num_pigs << " * " << wt_pig << " + "
                 << num_sheeps << " * " << wt_sheep << " = "
                 << total_wt << endl;
    }
    return 0;
}
```

Problems of Comparing Floating-point Numbers

```
#include <iostream>     /* File: float-comparison-v1.cpp */
using namespace std;

int main()
{
    float x = 0.1;
    float product = 10.0 * x;

    float sum = 0.0;
    for (int i = 0; i < 10; ++i)
        sum += x;

    cout << "sum = " << sum << endl;
    cout << "product = " << product << endl;
    cout << "10.0 * x = " << 10.0 * x << endl;
    cout << (sum == product) << endl;

    return 0;
}
```

Problems of Comparing Floating-point Numbers ..

```cpp
#include <iostream>     /* File: float-comparison.cpp */
using namespace std;

int main()
{
    float x = 0.1;
    float product = 10.0 * x;

    float sum = 0.0;
    for (int i = 0; i < 10; ++i)
        sum += x;

    // Set output precision to 10 significant figures
    cout.precision(10);
    // Print boolean outputs as true or false instead of 1 and 0
    cout << boolalpha;

    cout << "sum = " << sum << endl;
    cout << "product = " << product << endl;
    cout << "10.0 * x = " << 10.0 * x << endl;
    cout << (sum == product) << endl;
    return 0;
}
```

Count Animals: Further Work

- Further check ways to compare floating point numbers here.
- What if the total number of animals is not more than 10.
- What if we have three types of animal instead of two? (You may ignore the constraint on the total number of animals.)
- For those who know recursion, can you work out a recursive solution?

Part II

GPA Calculator

GPA Calculator

- Assume the following letter grade to grade point conversion:

Letter Grade	Grade Point
A	4.0
B	3.0
C	2.0
D	1.0
F	0.0

- Design a program that calculates a student's GPA (grade point average).

Typical Output

```
No.  of credits of your course (0 to stop):  3
Your letter grade (A, B, C, D or F): A
No.  of credits of your course (0 to stop):  4
Your letter grade (A, B, C, D or F): B
No.  of credits of your course (0 to stop):  2
Your letter grade (A, B, C, D or F): E
Invalid input, please enter your grade again!
No.  of credits of your course (0 to stop):  2
Your letter grade (A, B, C, D or F): D
No.  of credits of your course (0 to stop):  0
You have taken a total of 9 credits ...
and your GPA is 2.88889
```

Program Requirements

- A student first enters the number of credits of his/her course.
- The program stops if the number of credits is ≤ 0.
- The student then enters the letter grade A, B, C, D or F.
- Invalid letter grades are ignored and the student is prompted to re-enter the grade.
- The program shall calculate the total number of credits earned by the student and his/her GPA according to the following formula:

$$\frac{\sum_{i=1}^{n} (grade_i * credit_i)}{\sum_{i=1}^{n} credit_i}$$

Program Design

Major components of the program:

1. A loop for each taken course to
   - (a) ask for the number of credits
   - (b) ask for the letter grade
   - (c) convert the letter grade to grade points
   - (d) accumulate the total credits and grade points
2. Calculate and output the GPA.

You will need some variables to hold:

- the number of credits
- the letter grade
- the converted grade points
- the sum of credits
- the sum of grade points

Variables

```cpp
char grade;
/* Don't forget to initialize some of these variables */
int num_credits, total_num_credits = 0;
int total_grade_points = 0; // Is it a good idea to use integer type here?

// Think about the output
cout << "You have taken a total of " << total_num_credits
     << " credits ..." << endl << "and your GPA is "
     << total_grade_points / total_num_credits << endl;
```

Recall the usual arithmetic conversion rules for binary operations:

- If all operands are int
  - compute using integer arithmetic
- If one operand is double/float
  - convert the other operand to double/float
  - compute using floating-point arithmetic
  - return the result in double/float

GPA Calculator: Using if

```
char grade;
int num_credits, total_num_credits = 0;
double total_grade_points = 0;

cout << "No. of credits of your course (0 to stop): ";
cin >> num_credits;
cout << "Your letter grade (A, B, C, D or F): ";
cin >> grade;
total_num_credits += num_credits; // Update total no. of credits

if (grade == 'A')       // Convert letter grade to grade point
    total_grade_points += num_credits * 4;
else if (grade == 'B')
    total_grade_points += num_credits * 3;
else if (grade == 'C')
    total_grade_points += num_credits * 2;
else if (grade == 'D')
    total_grade_points += num_credits * 1;
else if (grade == 'F')
    total_grade_points += num_credits * 0;
else
    total_num_credits -= num_credits;
```

GPA Calculator: Using switch

```
/*
 * Codes for variables definition and initialization, and inputs
 */

switch (grade)          // Convert letter grade to grade point
{
    case 'A':
        total_grade_points += num_credits * 4; break;
    case 'B':
        total_grade_points += num_credits * 3; break;
    case 'C':
        total_grade_points += num_credits * 2; break;
    case 'D':
        total_grade_points += num_credits * 1; break;
    case 'F':
        total_grade_points += num_credits * 0; break;
    default:
        total_num_credits -= num_credits;
}
```

GPA Calculator: Using if .. Allowing Small Case

```
/*
 * Codes for variables definition and initialization, and inputs
 */

if ((grade == 'A') || (grade == 'a'))
    total_grade_points += num_credits * 4;

else if ((grade == 'B') || (grade == 'b'))
    total_grade_points += num_credits * 3;

else if ((grade == 'C') || (grade == 'c'))
    total_grade_points += num_credits * 2;

else if ((grade == 'D') || (grade == 'd'))
    total_grade_points += num_credits * 1;

else if ((grade == 'F') || (grade == 'f'))
    total_grade_points += num_credits * 0;

else
    total_num_credits -= num_credits;
```

GPA Calculator: Using switch .. Allowing Small Case

```
switch (grade)          // Convert letter grade to grade point
{
    case 'A':
    case 'a':
        total_grade_points += num_credits * 4; break;
    case 'B':
    case 'b':
        total_grade_points += num_credits * 3; break;
    case 'C':
    case 'c':
        total_grade_points += num_credits * 2; break;
    case 'D':
    case 'd':
        total_grade_points += num_credits * 1; break;
    case 'F':
    case 'f':
        total_grade_points += num_credits * 0; break;
    default:
        total_num_credits -= num_credits;
}
```

GPA Calculator: Complete Program .. Using constants

```cpp
#include <iostream>     /* File: gpa.cpp */
using namespace std;

const double A = 4.0;   // Definition of Constants
const double B = 3.0;
const double C = 2.0;
const double D = 1.0;
const double F = 0.0;

int main()
{   // Variables definition and initialization
    char grade;
    int num_credits, total_num_credits = 0;
    double total_grade_points = 0;

    do
    {
        cout << "No. of credits of your course (0 to stop): ";
        cin >> num_credits;

        if (num_credits <= 0) // What does this do?
            break;

        cout << "Your letter grade (A, B, C, D or F): ";
        cin >> grade;
        total_num_credits += num_credits; // Update total no. of credits

```

GPA Calculator: Complete Program .. Using constants ..

```
            switch (grade)  // Convert letter grade to grade point
            {
                case 'A':   // No break here; execute code in case 'a'
                case 'a':
                    total_grade_points += num_credits * A; break;
                case 'B':
                case 'b':
                    total_grade_points += num_credits * B; break;
                case 'C':
                case 'c':
                    total_grade_points += num_credits * C; break;
                case 'D':
                case 'd':
                    total_grade_points += num_credits * D; break;
                case 'F':
                case 'f':
                    total_grade_points += num_credits * F; break;
                default :
                    cout <<
                    ↪  "Invalid input, please enter your grade again!\n";
                    total_num_credits -= num_credits;
            }
        } while (true);     // Why is this not an infinite loop?
```

GPA Calculator: Complete Program .. Using constants ..

```

    cout << "You have taken a total of " << total_num_credits
         << " credits ..." << endl << "and your GPA is "
         << total_grade_points / total_num_credits << endl;

    return 0;
}
```