Function Basics

Week 3
2017 Fall

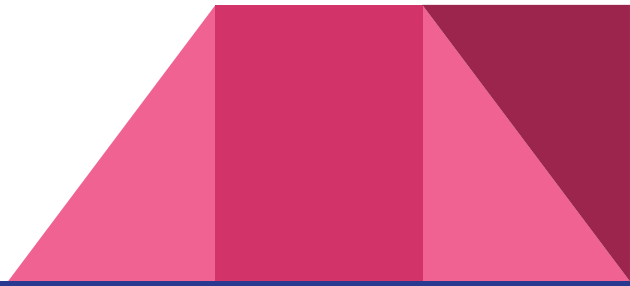
Computer Programming for Engineers

Cassiano Campes <cassianocampes@gmail.com>
Yoohyuk Lim <dburg3065@gmail.com>

Problem 1:
Let’s buy clothes

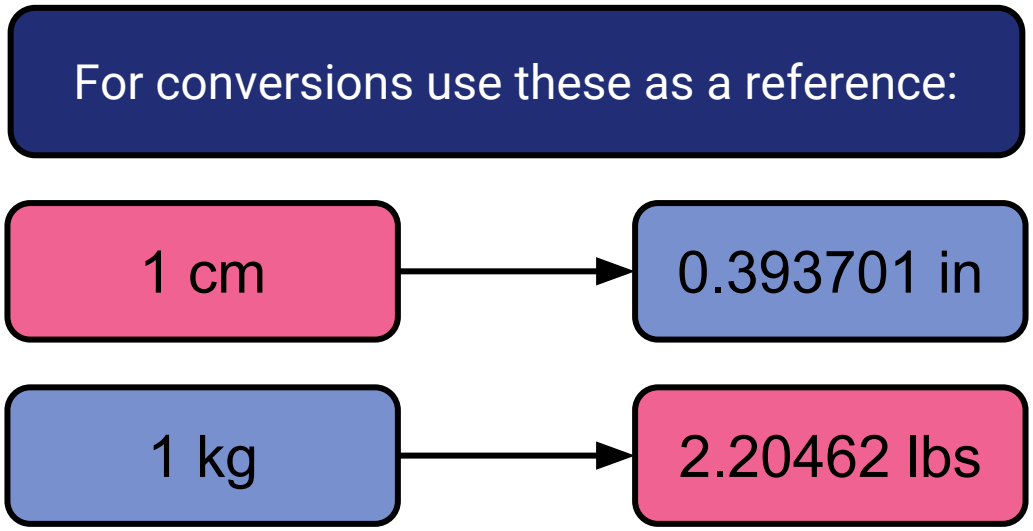
Problem 1: Let’s buy clothes (1/4)

- Write a program that asks users the following information:
  - Height, weight, and age (use proper variable types)
- These information are used to compute clothing size
- Use functions for each calculation



Problem 1: Let’s buy clothes (2/4)

- **Hat size** is equal to the weight **(in pounds),** divided by height **(in inches)** and thus all multiplied by 2.9



Problem 1: Let’s buy clothes (3/4)

- **Jacket size (in):** height times weight divided by 288
  - Then adjusted by adding one-eighth of an inch for each 10 years over age 30

Note that the adjustment only takes place after a full 10 years. So, there is no adjustment for ages 30 through 39, but one eighth of an inch is added for age 40 and so on.

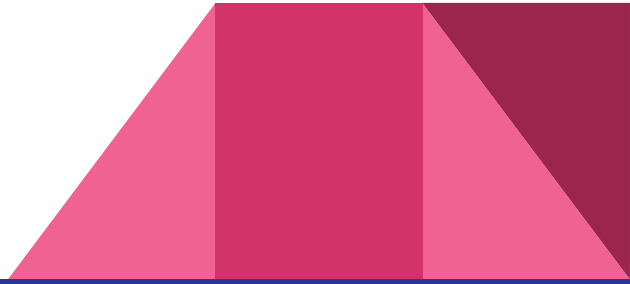
Problem 1: Let’s buy clothes (4/4)

- **Waist (in):** weight divided by 5.7 and then adjusted by adding one-tenth of an inch for each 2 years over 28

Note that the adjustment only takes place after a full 2 years. So, there is no adjustment for age 29, but one-tenth of an inch is added for age 30

Problem 1: Hints

- You probably need to create these functions:
  - Function **getHatSize(...);**
  - Function **getJacketSize(...);**
  - Function **getWaistSize(...);**
  - Function **mainMenu(...);**
- Don’t forget to use **static_cast<>** when needed



Problem 2:
Let’s win the lottery price!

Problem 2: Let’s win the lottery price!

- A typical 6/49 game, each player chooses six **non-duplicate** numbers from a range of 1-49
- If the six numbers on a ticket match the numbers drawn by the lottery, the ticket holder is a jackpot winner - **regardless of the order of the numbers**

The math behind this is the **Combination** formula

Problem 2: Let's win the lottery price!

- The Combination formula is:
  - **n =** the number of possible values to choose
  - **k =** the number of selected values

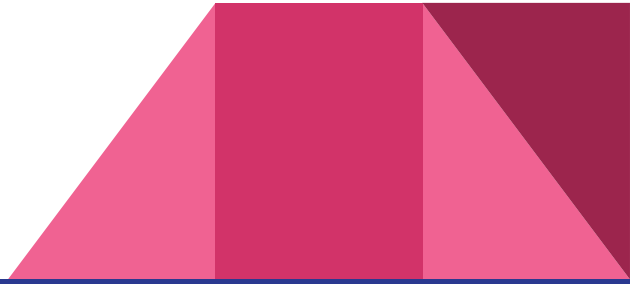
$$C(n,k) = \frac{n!}{(n-k)!k!}$$

- In Combination formula, does not matter the ordering

Problem 2: Hints

- Would be better to create first a function
  - **calcFactorial(...)** to calculate a single number
- Use this function to calculate the combination formula

Caution: If you use big numbers, stack overflow happens



Answer: Problem 1

Problem 1: Solution

```cpp
#include <iostream>
#include <math.h>
using namespace std;

float convertCmToIn(int cm)
{
    /* We need to cast the integer (cm) to return a float */
    return (static_cast<float>(cm) * 0.393701);
}

float convertKgToLbs(float weight)
{
    /* Here we don't need to cast because all variables are float*/
    return (weight * 2.20462);
}

float getHatSize(int height, float weight) {
    /* Here we need to use the convert functions
     * [height] is in centimeter -> converts to inches
     * [weight] is in kilograms -> converts to pounds
     * */
    return (convertKgToLbs(weight) / convertCmToIn(height) * 2.9);
}

```

Problem 1: Solution

```
25. float getJacketSize(int height, int age, float weight) {
26.     float result;
27.     float adjust = 0;
28.
29.     /* First we check if the age is above the adjustment condition
30.      * If so, we use modulus operator to get the remainder of the division
31.      *      and thus we can determine if we need to subtract or not
32.      *      the age to identify how many times we must add the adjustment.
33.      * Otherwise, the adjustment is left as 1, and does not affect the
34.      *      multiplication in the return statement of the function.
35.      **/
36.     if (age >= 30) {
37.         if ((age % 10) != 0)
38.             age = age - (age % 10);
39.         adjust = (1.0/8)*((age - 30) / 10);
40.     }
41.
42.     /* We convert the values and multiply accordingly to what is requested */
43.     result = ((convertCmToIn(height) * convertKgToLbs(weight)) / 288);
44.
45.     /* Returning the result from the function. Notice that we had previously
46.      *      calculated the adjust variable, that can be >= 1 */
47.     return (result + static_cast<float>(adjust));
48. }
49.
```

Problem 1: Solution

```
50. float getWaistSize(int age, float weight) {
51.     float result;
52.     float adjust = 0;
53.
54.     /* Same as the above mentioned about checking the age for the adjustment */
55.     if (age >= 28)
56.     {
57.         if ((age % 2) != 0)
58.             age = age - (age % 2);
59.         adjust = (1.0/10)*((age - 28) /2);
60.     }
61.
62.     /* We convert the value and multiply accordingly to what is requested */
63.     result = (convertKgToLbs(weight) / 5.7);
64.
65.     /* Returning the result from the function. Notice that we had previously
66.      *      calculated the adjust variable, that can be >= 1 */
67.     return (result +  static_cast<float>(adjust));
68. }
69.
70. int Menu() {
71.     /* Store inputs from the user */
72.     int height;
73.     int age;
74.     float weight;
75.
```

Problem 1: Solution

```
76.     /* Stores the output to be given to users */
77.     float hat_size;
78.     float jacket_size;
79.     float waist_size;
80.
81.     /* Let's show something to the user */
82.     cout << "Input your [height] in cm, [weight] in kg, and [age] in years:" << endl;
83.
84.     /* Get the inputs from the user, notice that we concatenate the variables,
85.      *      thus, the inputs must be separated by spaces */
86.     cin >> height >> weight >> age;
87.
88.     cout << "Hat size: " << getHatSize(height, weight) << endl;
89.     cout << "Jacket size: " << getJacketSize(height, age, weight) << endl;
90.     cout << "Waist size: " << getWaistSize(age, weight) << endl;
91. }
92.
93. int main() {
94.
95.     /* Set the format for the floating point variables */
96.     cout.setf(ios::fixed);
97.     cout.setf(ios::showpoint);
98.     cout.precision(2);
99.
```

Problem 1: Solution

```
100.    /* This is a auxiliary function called to simplify the main() function
101.     *      Menu() is in charge of calling all the other functions accordingly */
102.    Menu();
103.
104.    return 0;
105. }
```

Answer: Problem 2

Problem 2: Solution

```cpp
#include <iostream>
#include <stdio.h>
#include <math.h>
using namespace std;

/* This is the basic function that is used to calculate factorial
 * As you can notice, the function calls itself by decrementing
 *      its iteration index 'n'.
 *  Every time the function calls itself, it is decremented and
 *      once the index reach the value of 1, the function
 *      returns.
 *  Let's suppose we called calcFactorial(4), the behavior is
 *      as follows:
 *      calcFactorial(4)
 *      \--> calcFactorial(3)
 *           \--> calcFactorial(2)
 *                \--> calcFactorial(1)
 *
 *                At this point 1 is reached, then, the functions
 *                starts to return the value */
long calcFactorial(long n) {
    long tmp = 0;
    if (n <= 1) {
        return 1;
    }
```

Problem 2: Solution

```
26.     return (n * calcFactorial((n - 1)));
27. }
28.
29. /* This function calculates the combination formula
30.  * We use the calcFactorial primitive function to get
31.  * the calculation done */
32. long  calcCombination(long n, long k) {
33.
34.     return (calcFactorial(n) / (calcFactorial(k) * calcFactorial((n-k))));
35. }
36.
37. /*
38.  * This is the main function where we get the input from the user
39.  *      about the combination to be calculated */
40. int main() {
41.     long n, k;
42.     cout << "Enter the [n] and the [k] combination values: " << endl;
43.     cin >> n >> k;
44.
45.     cout << calcCombination(n, k) << endl;
46.
47.     return 0;
48. }
```