2.10 — Chapter 2 comprehensive quiz

BY ALEX ON JUNE 11TH, 2007 | LAST MODIFIED BY ALEX ON OCTOBER 24TH, 2017

Quick Review

Integers are used for holding whole numbers. When using integers, keep an eye out for overflow and integer division problems. Use the int type when the size of an integer doesn't matter. Use fixed-width integers when the precise size of an integer is important (either due to range or memory usage concerns).

Floating point numbers are used for holding real numbers (which can have fractional components). When using floating point numbers, keep an eye out for precision issues, rounding errors, and comparison issues.

Boolean values hold only true and false. They do not have any major issues.

Char values are integers that can be interpreted as an ASCII value. When using chars, be careful not to mix up ASCII code values and numbers, and watch for overflow and integer division problems.

Use the const keyword to declare symbolic constants instead of #define. It's safer.

Comprehensive quiz

Question 1

Why are symbolic constants usually a better choice than literal constants? Why are const symbolic constants usually a better choice than #defined symbolic constants?

Hide Solution

Using literal constants (aka. magic numbers) in your program makes your program harder to understand and harder to modify. Symbolic constants help document what the numbers actually represent, and changing a symbolic constant at it's declaration changes the value everywhere it is used. #define constants do not show up in the debugger and are more likely to have naming conflicts.

Question 2

Pick the appropriate data type for a variable in each of the following situations. Be as specific as possible. If the answer is an integer, pick either int, long, or a a specific fixed-width integer type (e.g. int16_t) based on range. If the variable should be const, say so.

a) The age of the user (in years)
b) Whether the user wants color or not
c) pi (3.14159265)
d) The number of pages in a textbook (assume size is important)
e) The price of a stock in dollars (to 2 decimal places)
f) How many times you've blinked since you were born (note: answer is in the millions)
g) A user selecting an option from a menu by letter
h) The year someone was born (assuming size is important)

Hide Solution

a) int
b) bool
c) const double
d) Since books can often have more than 255 pages but probably never have more than 32,767 pages, int16_t is a good choice here.
e) float
f) long
g) char
h) int16_t. You can use positive numbers to represent AD birthdates, and negative numbers to represent BC birthdates.

Question 3

Write the following program: The user is asked to enter 2 floating point numbers (use doubles). The user is then asked to enter one of the following mathematical symbols: +, -, *, or /. The program computes the answer on the two numbers the user entered and prints the results. If the user enters an invalid symbol, the program should print nothing.

Example of program:

```
Enter a double value: 6.2
Enter a double value: 5
Enter one of the following: +, -, *, or /: *
6.2 * 5 is 31
```

Hint: You can check if the user has entered a plus symbol using an if statement, covered in section **2.6 -- Boolean values**.

Hide Solution

```cpp
#include <iostream>

double getDouble()
{
    std::cout << "Enter a double value: ";
    double x;
    std::cin >> x;
    return x;
}

char getOperator()
{
    std::cout << "Enter one of the following: +, -, *, or / ";
    char op;
    std::cin >> op;
    return op;
}

void printResult(double x, char op, double y)
{
    if (op == '+')
        std::cout << x << " + " << y << " is " << x + y << '\n';
    else if (op == '-')
        std::cout << x << " - " << y << " is " << x - y << '\n';
    else if (op == '*')
        std::cout << x << " * " << y << " is " << x * y << '\n';
    else if (op == '/')
        std::cout << x << " / " << y << " is " << x / y << '\n';
}

int main()
{
    double x = getDouble();
    double y = getDouble();

    char op = getOperator();

    printResult(x, op, y);

    return 0;
}
```

Question 4

This one is a little more challenging. Write a short program to simulate a ball being dropped off of a tower. To start, the user should be asked for the initial height of the tower in meters. Assume normal gravity (9.8 m/s^2), and that the ball has no initial velocity (the ball is not moving to start). Have the program output the height of the ball above the ground after 0, 1, 2, 3, 4, and 5 seconds. The ball should not go underneath the ground (height 0).

Your program should include a header file named constants.h that includes a namespace called myConstants. In the myConstants namespace, define a symbolic constant to hold the value of gravity (9.8). See section **2.9 -- Symbolic constants and the const keyword** for a reminder on how to do this.

Use a function to calculate the height of the ball after x seconds. The function can calculate how far the ball has fallen after x seconds using the following formula: distance fallen = gravity_constant * x_seconds^2 / 2

Sample output:

```
Enter the initial height of the tower in meters: 100
At 0 seconds, the ball is at height: 100 meters
At 1 seconds, the ball is at height: 95.1 meters
At 2 seconds, the ball is at height: 80.4 meters
At 3 seconds, the ball is at height: 55.9 meters
At 4 seconds, the ball is at height: 21.6 meters
At 5 seconds, the ball is on the ground.
```

Note: Depending on the initial height, the ball may not reach the ground in 5 seconds -- that’s okay. We’ll improve this program once we’ve covered loops.
Note: The ^ symbol isn’t an exponent in C++. Implement the formula using multiplication instead of exponentiation.

Hide Solution

In constants.h:

```cpp
#ifndef CONSTANTS_H
#define CONSTANTS_H

namespace myConstants
{
    const double gravity(9.8); // in meters/second squared
}
#endif
```

In your main code file:

```cpp
#include <iostream>
#include "constants.h"

// gets initial height from user and returns it
double getInitialHeight()
{
    std::cout << "Enter the height of the tower in meters: ";
    double initialHeight;
    std::cin >> initialHeight;
    return initialHeight;
}

// Returns height from ground after "seconds" seconds
double calculateHeight(double initialHeight, int seconds)
{
    // Using formula: [ s = u * t + (a * t^2) / 2 ], here u(initial velocity) = 0
    double distanceFallen = (myConstants::gravity * (seconds * seconds)) / 2;
    double currentHeight = initialHeight - distanceFallen;

    return currentHeight;
}

// Prints height every second till ball has reached the ground
void printHeight(double height, int seconds)
{
    if (height > 0.0)
        std::cout << "At " << seconds << " seconds, the ball is at height: " << height << " meters\n";
    else
        std::cout << "At " << seconds << " seconds, the ball is on the ground.\n";
}

void calculateAndPrintHeight(double initialHeight, int seconds)
{
    double height = calculateHeight(initialHeight, seconds);
    printHeight(height, seconds);
}

int main()
{
    const double initialHeight = getInitialHeight();

    calculateAndPrintHeight(initialHeight, 0);
    calculateAndPrintHeight(initialHeight, 1);
    calculateAndPrintHeight(initialHeight, 2);
    calculateAndPrintHeight(initialHeight, 3);
    calculateAndPrintHeight(initialHeight, 4);
    calculateAndPrintHeight(initialHeight, 5);

    return 0;
}
```

Note that calculateHeight() doesn't print the height itself, under the best practice that functions should do one and only one thing. We use a different function to do the printing.

3.1 -- Operator precedence and associativity

Index

2.9 -- Const, constexpr, and symbolic constants

Share this:

C++ TUTORIAL | PRINT THIS POST

530 comments to 2.10 — Chapter 2 comprehensive quiz

« Older Comments 1 … 6 7 8

Matt

July 16, 2018 at 7:54 am · Reply

Hello, for question 4,I'm getting slightly different answers than the example output given:

The sample output for 100m:
Enter the initial height of the tower in meters: 100
At 0 seconds, the ball is at height: 100 meters
At 1 seconds, the ball is at height: 95.1 meters
At 2 seconds, the ball is at height: 80.4 meters
At 3 seconds, the ball is at height: 55.9 meters
At 4 seconds, the ball is at height: 21.6 meters
At 5 seconds, the ball is on the ground.

My output:
Enter tower height in metres: 100