

Feb 11, 09 7:27

private_vs_public.cpp

Page 1/1

```

// file: private_vs_public.cpp
//
// Self-contained test file for private vs. public variables and
// functions in C++ classes.
//
// Programmer: Dick Furnstahl  furnstahl.1@osu.edu
//
// Revision history:
//   02/10/09  original version
//
// Notes:
//   * For compactness, we include the class prototype and
//     class definition in this file at the top
//   * We've implicitly used the default destructor by not including it.
//   * Short functions can be declared "inline" in the prototype.
//   * Note the use of a "get" method to access the value of a private
//     variable from outside the class.
//
//*****
// include files
#include <iostream>    // cout and cin
#include <iomanip>     // manipulators like setprecision

class PrivacyTest
{
public:
    PrivacyTest (double value_passed); // constructor
    double xsq () {return (x*x)};    // a public function
    double x;                       // a public variable
    double get_y () {return (y)};    // a public "get" function

private:
    double ysq () {return (y*y)};    // a private function
    double y;                       // a private variable
};

PrivacyTest::PrivacyTest (double value_passed)
{
    x = value_passed;    // set an internal variable to a passed value
    y = value_passed;    // set an internal variable to a passed value
}

//*****

int main ()
{
    double value_passed = 10.;
    std::cout << "Original value is " << value_passed << std::endl;

    PrivacyTest my_PrivacyTest (value_passed);    // create a PrivacyTest object

    // Try getting the value of x in the object
    std::cout << "Public x: " << my_PrivacyTest.x << std::endl;
    std::cout << "Public function for x^2: "
        << my_PrivacyTest.xsq() << std::endl;

    // Try changing the value of x in the object and printing again
    my_PrivacyTest.x = 20.;
    std::cout << "Public x is now: " << my_PrivacyTest.x << std::endl;

    // Get and print the private value of
    std::cout << "Private y is now: " << my_PrivacyTest.get_y()
        << std::endl;
}

//*****

```