UNIT TESTING FOR FRAMEWORK PROJECT

# ORIENTED UNIT TESTING FOR FRAMEWORK PROJECT

* Keep testing at unit level: Designing a test class for each normal class and function of the class should be tested separately.
* Keep all test cases in the unit test project independently executed: each test case does not depend on other tests and does not depend on the order in which the tests are executed.
* Cover boundary cases for function test.
* Testing normal conditions.
* Testing unexpected conditions.
* Bad input values.
* Boundary conditions.
* Measure the tests: Apply coverage analysis to the test runs so that it is possible to read the exact execution coverage and investigate which parts of the code is executed and not.

# TEST CASE FOR “ELLIPSE.CPP”

# Test plan

* Using Google Test framework.
* Designing test class name: EllipseTest.cpp.
* Understand the requirements of the methods in “Ellipse” class.

Ellipse class with:

* Methods of class:

Ellipse(void);

Ellipse(const Ellipse &ell);

Ellipse(int cx, int cy, int a, int b, double f);

virtual void get(int &cx, int &cy, int &a, int &b, double &f) const;

virtual ~Ellipse();

* Members of class: data type “int” and “double”
* Write test cases for each method in the class.

# Test analysis

1. Create test cases for default constructor method.
2. Create test cases for copy constructor method.
3. Create test cases for constructor with parameters method.
4. Create test cases for “get” method.

# Design test case

* 1. Test case of default constructor.

Purpose: setting initial values for certain member variables

* Create some test cases with “DefaultCtor” name.
* Input: any value of parameters to satisfying boundary conditions of data type are integer and double (negative value, 0, positive value).
* Call default constructor method.
* Using “get” function to check the result of ellipse information.
* Expected result same requirement or not
  1. Test case of copy constructor method

Purpose: new object is created from an existing object.

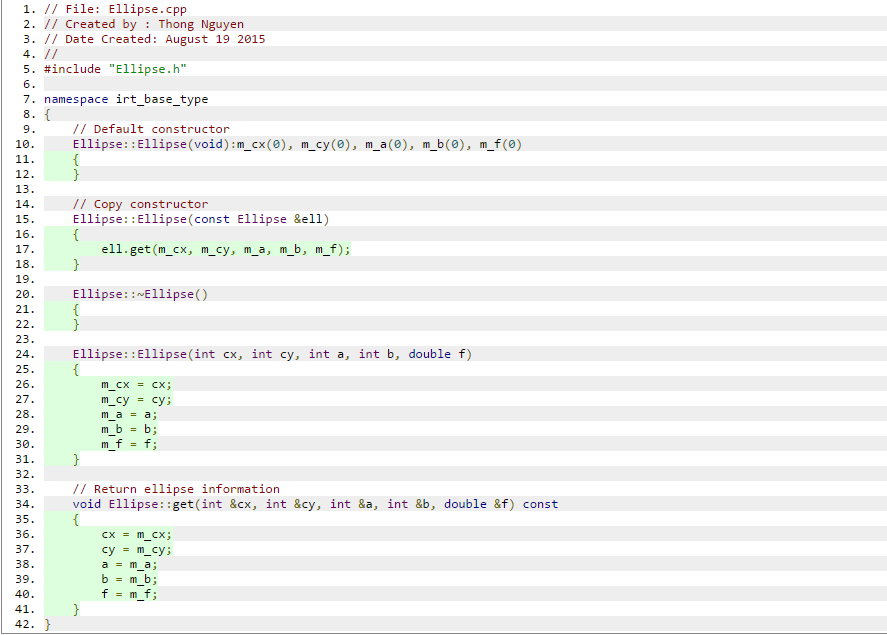
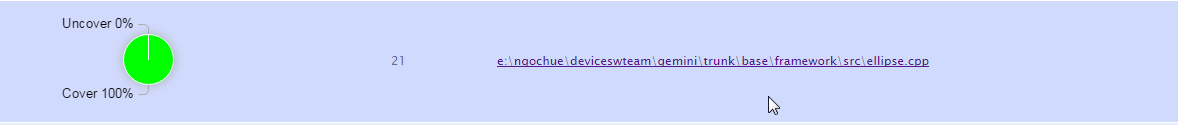
* Create some test cases with “CopyCtor” name.
* Input: Create one object ellipse
* Call copy constructor method.
* Using “get” method to check the result of ellipse information.
* Expected result same with input object or not.
  1. Test case of constructor with parameters method

Purpose: setting initial values for certain member variables from input value

* Create some test cases with :Ctor” name
* Input: any value of parameters to satisfying boundary conditions of data type are integer and double (negative value, 0, positive value).
* Call constructor with parameters method
* Using “get” method to check result.
* Expected result: same with input values or not.
  1. Test case of “get” method
* Create test case with “Get” name
* Input: any value of parameters to satisfying boundary conditions of data type are integer and double (negative value, 0, positive value)
* Using constructor default, copy constructor, constructor with parameters to initial values for certain member variables.
* Call “get” method.
* Expected result: same with input values or not.

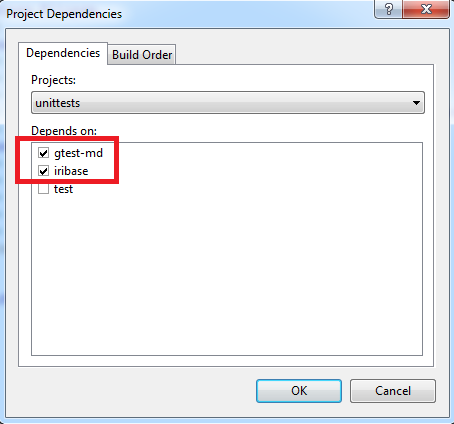
# CHECK TEST CODE COVERAGE FOR “ELLIPSE.CPP”

# Using OpenCppCoverage to check test code coverage

Cover 100%

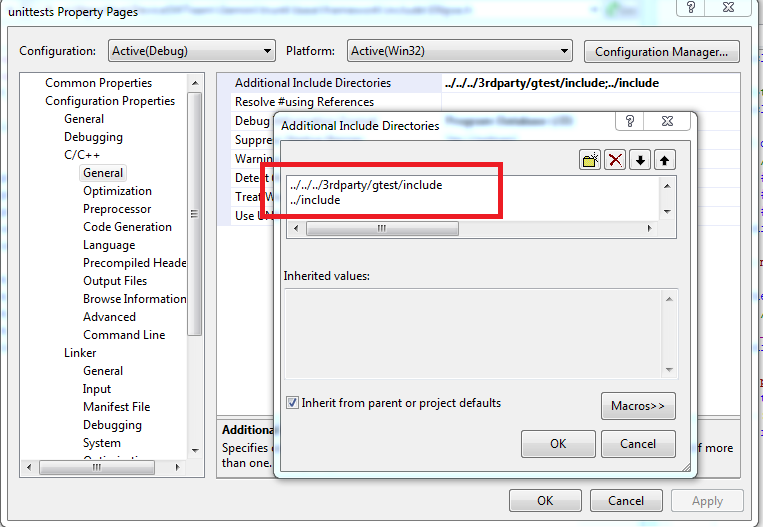
# SET UP PROJECT FRAMEWORK WITH GOOGLE TEST

* Include gtest-md.vcproj library.
* Project need test name: iribase.
* Step 1: Create project name: “unittests” to write unit test for all class for iribase project.
* Step 2: select “Project dependencies…



* Step 3: Select “Configuration Properties”

Add include directories:



* Step 4: Select “Runtime library” is “Multi-threaded Debug DLL(/MTd)”
* Step 5: Set as Startup project and build project.
* Step6: Can filter with test suite or test name in main function of file unittests.cpp.

Ex:



Result show:

