Unit Testing
with the
Boost Testing Framework

Boost Testing Framework

- The Unit Test Framework is responsible for supplying function main() that initializes the testing environment and taking care about results reporting.

Setting up the Testing Framework

- Add *boost_unit_test_framework* library
- #include <boost/test/unit_test.hpp>
- #define BOOST_TEST_DYN_LINK
  - Tells the compiler to use dynamic linking
- #define BOOST_TEST_MODULE TestSuite1
  - Sets up a test suite called “TestSuite1”
- using namespace boost::unit_test

Adding Test Cases

- The easiest way to register a test case is by using Boost's macros

BOOST_AUTO_TEST_CASE

Adding Test Cases

```
// Test case called “test1”
BOOST_AUTO_TEST_CASE( test1 )
{
    // Test body
}
```

Macros for Assertions

- BOOST_CHECK(param);
  - Asserts that param is *true*
- BOOST_CHECK_EQUAL(a, b);
  - Asserts that *a* == *b*
- *_MESSAGE(…);
  - Works the same way as the normal versions, but uses the provided message instead of the default failure message.

Macros for Assertions

- BOOST_REQUIRE(predicate);
  - Aborts the current test case if predicate is *false*
- BOOST_ERROR(message);
  - Unconditional error. Useful inside of if statements.
- BOOST_FAIL(message);
  - Unconditional failure. Similar to BOOST_ERROR, but aborts the test case.