National University of computer and emerging sciences

LAB 01

Summary

Items	Description
Course Title	Object Oriented Programming
Lab Title	Introduction to Debugging and Testing
Duration	3 Hours
Operating System/Tool/ Language	Ubuntu/ Eclipse, Gtest / C++
Objective	To get familiar with the Eclipse and Gtest

Installation of Eclipse

Please follow this tutorial to install Eclipse:

Guide to install Eclipse

Installation of Google Test (Gtest)

1. Download the file install-libraries.sh from Google Classroom.
2. Open the terminal and change the directory to current folder (where you have downloaded the file).
3. Run the following command
   bash install-libraries.sh

Now, we are good to go.

Debugging the Code using Eclipse:

You need to perform the following to complete the task.

1. Download Exercise.cpp file from Google Classroom.
2. Now open eclipse and create a new C++ project.
3. Name your project as DebugExercise.

4. Copy code from Exercise.cpp file and paste it in your new project file.

5. Now add breakpoints in the code at mentioned line.

6. For example in the following figure, a code snippet in which breakpoints are inserted at line 13.

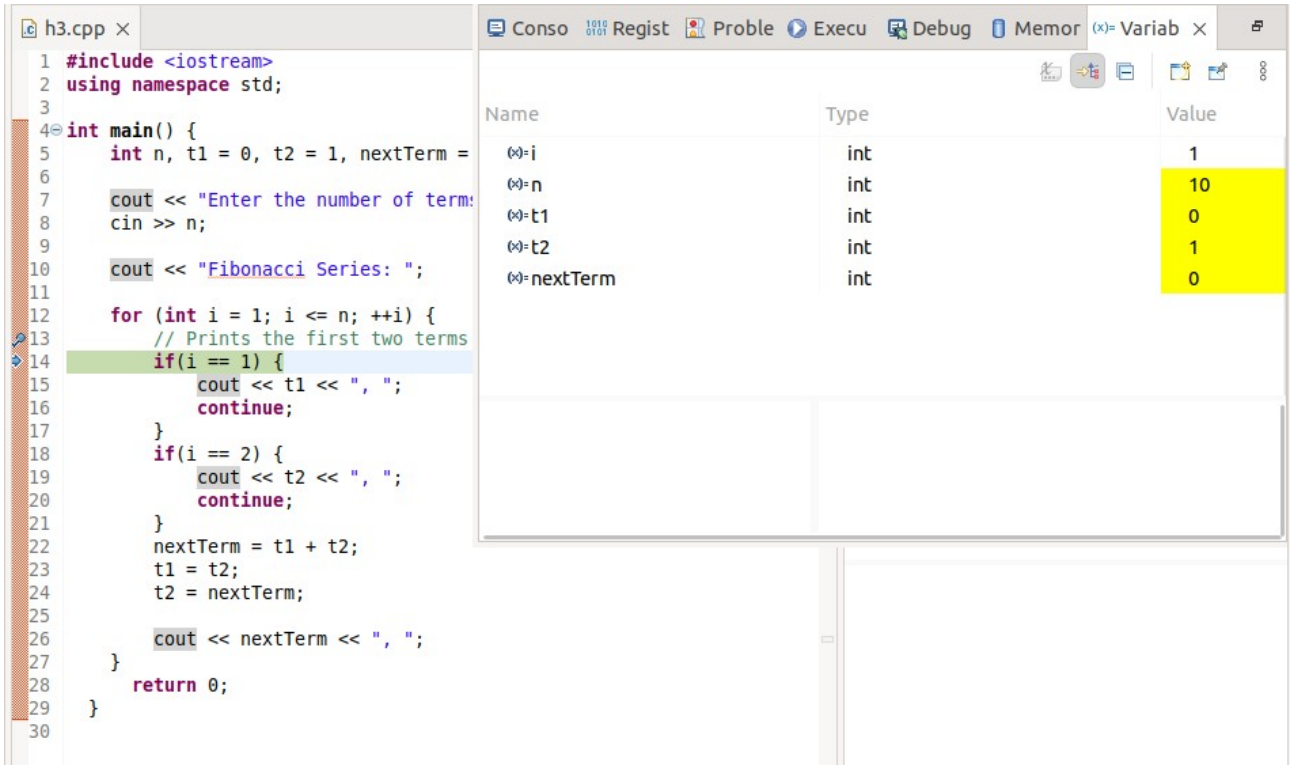
```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n, t1 = 0, t2 = 1, nextTerm = 0;
6
7      cout << "Enter the number of terms: ";
8      cin >> n;
9
10     cout << "Fibonacci Series: ";
11
12     for (int i = 1; i <= n; ++i) {
13         // Prints the first two terms.
14         if(i == 1) {
15             cout << t1 << ", ";
16             continue;
17         }
18         if(i == 2) {
19             cout << t2 << ", ";
20             continue;
21         }
22         nextTerm = t1 + t2;
23         t1 = t2;
24         t2 = nextTerm;
25
26         cout << nextTerm << ", ";
27     }
28       return 0;
29   }
```

7. After adding breakpoints, start debugging the project by clicking on Debug Icon in the Quick Access toolbar.

8. Use step in and step out to check the value of each declared variable inside all functions.

9. For example in the following figure, the debugger stops at breakpoint inserted at line 13, and the values of all variables are shown in variable window.

National University
of computer and emerging sciences



Test using Gtest:

1. Create a file test.cpp and paste following code:

```
#include "What_to_test.cpp"
#include <gtest/gtest.h>

TEST(SquareRootTest, PositiveNos) {
   ASSERT_EQ(6.0, squareRoot(35.0));
   ASSERT_EQ(18.0, squareRoot(324.0));
   ASSERT_EQ(25.4, squareRoot(645.16));
   ASSERT_EQ(0, squareRoot(0.0));
}

TEST(SquareRootTest, NegativeNos) {
   ASSERT_EQ(-1.0, squareRoot(-15.0));
   ASSERT_EQ(-1.0, squareRoot(-0.2));
}

int main(int argc, char **argv) {
   testing::InitGoogleTest(&argc, argv);
   return RUN_ALL_TESTS();
}
```

2. Create another file named whattotest.cpp and paste following code (the function that we want to test in the test case):

```
#include <math.h>

double squareRoot(const double a) {
   double b = sqrt(a);
   if(b != b) { // nan check
      return -1.0;
   }else{
      return sqrt(a);
   }
}
```

3. After successful installation of libraries, you can run the tests.cpp by the following g++ command.
   g++ tests.cpp -lgtest -lpthread -o test

4. This will create the test executable file in the folder. Now you can run the test file by issuing a command ./test

   This will show the summary of the test cases which are passed and which are failed.

```
ansa@ansa-Vostro-14-3468:~$ ./test
[==========] Running 3 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 3 tests from TestName
[ RUN      ] TestName.Subtest_1
[       OK ] TestName.Subtest_1 (0 ms)
[ RUN      ] TestName.Subtest_2
test.cpp:10: Failure
Expected equality of these values:
  1
  2
[  FAILED  ] TestName.Subtest_2 (0 ms)
[ RUN      ] TestName.Subtest_3
[       OK ] TestName.Subtest_3 (0 ms)
[----------] 3 tests from TestName (0 ms total)

[----------] Global test environment tear-down
[==========] 3 tests from 1 test suite ran. (0 ms total)
[  PASSED  ] 2 tests.
[  FAILED  ] 1 test, listed below:
[  FAILED  ] TestName.Subtest_2

 1 FAILED TEST
```

National University
of computer and emerging sciences

Lab Tasks

Task#01

Write a C++ program to print the following:

Allah, *subhanahu wa ta'ala*, says:

> (10:106) Do not call upon any apart from Allah on those who have no power to benefit or hurt you. For if you call upon others than Allah you will be reckoned among the wrong-doers.

Task#02

Run the sample program and attach the output screenshots.

Task#03

Take an array of length n where all the numbers are nonnegative and unique. Find the element in the array possessing the highest value. Split the element into two parts where first part contains the lowest value in the array and second part hold the required additive entity to get the highest value. Return the array index value using a function named ‘splitBig( )’.
Input: 4 8 6 3 2
Modified Array: 4 2 6 6 3 2 (display the modified array)
Output: 1
use index number to return a value from splitBig().
Note: Download lab1.cpp and test1.cpp from googleClassroom.

Submission instructions:

1. Save all .cpp files with your roll no and task number
   e.g. i21XXXX_Task01.cpp
2. Save all screenshots of terminal with your roll no and task number
3. Now create a new folder with name ROLLNO_LAB03 e.g. i21XXXX_LAB03
4. Move all your .cpp files to this newly created directory and compress it into .zip file.
5. Now you must submit this zipped file on Google Classroom.

Good Luck