

CS150 Debugger Example

We are going to begin using the Visual Studio 2010 debugger to aid with debugging programs.

1. Go to **CS150-02 Public** folder and drag a copy of the solution DebugEx to your desktop.
2. Build your solution and then execute the program using the command "Start Without Debugging." You will see the following.

A screenshot of a Windows command prompt window. The title bar reads "C:\WINDOWS\system32\cmd.exe". The window contains the following text:

```
Simple Arithmetic Example
num1 / num2 is 0.833333
num4 / num5 is 0
Example Complete
Press any key to continue . . .
```

3. In looking at your output you wonder why num4 / num5 is 0 since both numbers are nonzero. What is going on?
4. Set a breakpoint by clicking an executable statement in the light blue column as follows. Notice the red button next to the executable statement cout.

A screenshot of the Visual Studio 2010 IDE showing a C++ source file named "main.cpp". The code is as follows:

```
1 // Debugger Example
2
3 #include <iostream>
4
5 using namespace std;
6
7 int main ()
8 {
9     double num1 = 5.0,
10         num2 = 6.0,
11         num3;
12     int num4 = 2,
13         num5 = 3,
14         num6;
15
16     cout << "Simple Arithmetic Example" << endl << endl;
```

A red circle breakpoint is set on line 16. The left margin shows a light blue column for line numbers and a red button for the breakpoint.

5. Let's start the debugger and step through our program an executable statement at a time. Start the debugger by going to Debug Start Debugging. The debugger will stop at the first executable statement that has a breakpoint. Your screen should look something like the following.

Make sure the Locals tab is selected and not the Autos tab.

The screenshot shows a C++ IDE with a debugger. The main window displays the following code:

```
1 // Debugger Example
2
3 #include <iostream>
4
5 using namespace std;
6
7 int main ()
8 {
9     double num1 = 5.0,
10         num2 = 6.0,
11         num3;
12     int num4 = 2,
13         num5 = 3,
14         num6;
15
16     cout << "Simple Arithmetic Example" << endl << endl;
17
18     num3 = num1 / num2;
19     num6 = num4 / num5;
20
21     cout << "num1 / num2 is " << num3 << endl;

```

The debugger is paused at line 16. The Locals window shows the following variables and their values:

| Name | Value | Type |
|------|--------------------------|--------|
| num4 | 2 | int |
| num1 | 5.0000000000000000 | double |
| num5 | 3 | int |
| num2 | 6.0000000000000000 | double |
| num6 | -858993460 | int |
| num3 | -9.2559631349317831e+061 | double |

The Call Stack window shows the following frames:

| Name | Lang |
|--|------|
| DebuggerEx.exe!main() Line 16 | C++ |
| DebuggerEx.exe!__tmainCRTStartup() Line 555 + | C |
| DebuggerEx.exe!mainCRTStartup() Line 371 | C |
| kernel32.dll!7c817077() | |
| [Frames below may be incorrect and/or missing, r | |

6. Show your instructor/TA the above screen.

- 7. Let's talk about what we are looking at.
- 8. From here we can continue running the program until the next breakpoint by selecting Resume as shown below.
- 9. You can also just execute the next statement by selecting Step Over.
- 10. We will talk about Step Into and Step Out when we get to functions.
- 11. Execute each statement up to but not including the return statement.

12. Show your instructor/TA

