Debug the World
an Introduction to Debugging Techniques
CSE Club Workshop
Overview

- What is Debugging?
- Writing Error-Resistant Code
- Understanding gcc/g++ Compiler Messages
- 7 Steps to Debug Your Code
- Debugging with Tools
MY CODE DOESN'T WORK
I HAVE NO IDEA WHY
MY CODE WORKS
I HAVE NO IDEA WHY
Writing Error-Resistant Code: Planning

Test Cases:
5 meter = 196.85 in
2.2 meter = 86.6142 in

Inputs:
length in meters

Output:
total feet, remaining inches

Pseudocode:
1) Ask for length in meters
2) Calculate
   total inches = meters * 39.3701
   total feet = total inches / 12 without remainder
   remaining inches = total inches - total feet * 12
3) Print the Answer
Writing Error-Resistant Code: Formatting

```cpp
}  
break;
}//ones
default:
{
    std::cout << "The conversion has failed." << std::endl;
    return PROGRAM_ERROR;
}
}//end conversion

//check for empty output
if(result.length() == EMPTY_OUTPUT)
{
    std::cout << "The entered number is invalid. \nPlease try again.\n";
}
else
{
    //print converted integer
    std::cout << "This is your number in roman numerals: ";
}
return 0;
}//main
```
Understanding Compiler Messages

fileName:lineNumber:columnNumber: error message

examples:
hello.cpp:6:2: error: expected ‘;’ before ‘return’
hello.cpp:5:2: error: ‘cout’ was not declared in this scope
cppfile.cpp: In function `int main()':
cppfile.cpp:6: no match for `_IO_ostream_withassign & >> char[7]'

Pertinent Code:

```cpp
cout >> "Hello!" >> endl;
```
cppfile.cpp: In function `int main()':
cppfile.cpp:7: parse error before `>'

Pertinent Code:

```cpp
cout << "Hello!" << endl
cin >> fValue;
```
In file included from cppfile.cpp:3:
Person.h:8: parse error before `:'
cppfile.cpp:8: syntax error before `<'
cppfile.cpp:9: syntax error before `>'

**Pertinent Code:**

In cppfile.cpp:

```cpp
#include "Person.h"

int main(void)
{
    float fValue;
    cout << "Hello!" << endl;
    cin >> fValue;
}
```

In Person.h:

```cpp
struct Person
{
    string strFirstName;
    string strLastName;
};
```
cppfile.cpp: In function `int main()':
cppfile.cpp:8: no match for `_IO_istream_withassign & >> Person &'
/afs/umd.edu/software/egcs/solaris/include/g++/iostream:223: candidates are: istream::operator
>>(streambuf *)
/afs/umd.edu/software/egcs/solaris/include/g++/iostream:222: istream::operator>>(istream &
(*)(istream &))
/afs/umd.edu/software/egcs/solaris/include/g++/iostream:221: istream::operator>>(ios & (*)(ios &))
/afs/umd.edu/software/egcs/solaris/include/g++/iostream:220: istream::operator>>(long double &)
/afs/umd.edu/software/egcs/solaris/include/g++/iostream:219: istream...
...
...
...
Pertinent Code:

#include <iostream>
#include "Person.h"

int main(void)
{
    Person p;
    cin >> p;
cppfile.cpp: In function `int main()':
cppfile.cpp:3: too few arguments to function `void myFunction(short int)'
cppfile.cpp:7: at this point in file

Pertinent Code:

```cpp
#include <iostream>

void myFunction(const short knNum);

int main(void)
{
    myFunction();
    return 0;
}
```
7 Steps to Debug Efficiently and Effectively

1. Always reproduce the bug before you change any code
2. Understand stack traces (the error messages)
3. Write a testcase that reproduces the bug
4. Know your error codes
5. Google! Bing! Duck! Duck! Go!
6. Pair Program your way out of it (an extra set of eyes)
7. Celebrate your fix
Tools

GDB
Visual Studio
VIM
Sublime Text Editor
Notepad++
https://goo.gl/5ulYU7