Department of Mathematics and Computing Science
CSCI 3430.1 - Principles of Programming Languages

Assignment #10 (the last assignment - at last)

Assigned : 17 November, 2013
Due : 24 November, 2013

Please add all source code into one big zip file and post that to Moodle, it won't take more than **ONE** file. If it's too large to upload to Moodle, you likely have more than source code. If you are still stuck and can't upload the file, email it to me, cc to yourself then see me in class.

1. (6 pts) Write code that demonstrates one computer language calling and receiving data back from another language. The first language reads two integers from the user (A, B) and passes them as binary values to a separately compiled function written in a different language. That second function divides the second number into the first (A/B) and returns the result to the caller as an integer. That first language then prints the integer it received from the second language out to screen. Again, the caller must be one language, the receiver a different one. You can use any method of inter-language linking or inter-process communications (**CORBA**, **ZMQ**, **DCOM, Twisted, Node.JS, dynamic linking**. **static linking**, RPC, etc) but the two parts must be distinctly different and both supplied (for example **VB calling C++**, **Python calling C, C# calling C++**, C calling Assembler, Java calling Pascal, etc). Please avoid the simple HTML calling javascript (that's just too easy). Please identify which languages, which versions of the compiler and which computer operating systems or environments you are using.

2. (2 pts) Describe (in your opinion) the best computer language for developing the following programs. Please state some sort of reason why you chose that language.
   a) App for the latest Apple iPod, iPad and iPhone
   b) Ubuntu Linux Gnome desktop GUI application
   d) Windows 8 desktop GUI application
   c) Android tablet Bluetooth keyboard device driver

3. (2 pts) Describe any two language defects that a hacker can use to create an exploit. For example, SQL ";", binary HTML/XML injection, late-bound substitution, C string buffer overflow, binary fuzzing, .. etc. Code is NOT necessary and definitely NOT recommended.