**Student Name:** **Student ID:**

# Objectives

* Explain the concept of “fuzzing.”
* Distinguish bug classes.
* Write code to implement buffer overflows.
* Use integer overflows.
* Use format string attacks.
* Create several types of fuzzers.

# Important Information

* For *every* lab and home assignment, store all your work in your personal repository in a subdirectory named **mXX**, where XX is the module number. Carefully name the program as described in each problem.
* Your programs are extracted from your repository by a Python script. If there are any errors in the program name, then your instructor will never see your program, and you will receive a mark of zero.
* Push your work to the server often, and ensure that you push the final version of a program by the deadline specified, because the script extracting them can be run at any time after the deadline.

# Instructions

1. Read chapter 8 in the *Gray Hat Python* textbook. The following links are also useful:
   * <https://www.sans.org/reading-room/whitepapers/malicious/basic-reverse-engineering-immunity-debugger-36982>
   * <https://sgros-students.blogspot.ca/2014/05/immunity-debugger-basics-part-1.html>
2. Complete Problem 5.

**Note:** Problem numbering continues from the module’s lab.

## Problem 5

1. Read chapter 8.2 in the *Grey Hat Python* textbook and create a program named **file\_fuzzer.py** using the snippets in the chapter.
2. Test your fuzzer and report the steps you took to make the fuzzer run.
3. Run the fuzzer against the standard Notebook application in Windows and report any crashes.