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

# Objectives

* Explain general-purpose CPU registers.
* Explain stack.
* Explain debug events.
* Explain soft breakpoints.
* Explain hardware breakpoints.
* Explain memory access faults.

# 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 chapters 1 and 2 in the *Gray Hat Python* textbook. The following links are also useful:
   * <https://docs.python.org/3/library/pdb.html>
   * <http://www.gnu.org/software/gdb/documentation/>
   * <http://sourceware.org/gdb/current/onlinedocs/gdb.pdf.gz>
   * <https://docs.python.org/2/extending/extending.html>
2. Complete Problems 4 and 5.

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

## Problem 4

1. Modify a program by renaming it **m05p03.c**, so that it reliably crashes and generates a core dump.
2. Perform a postmortem analysis using GDB and determine the locations of the stack frames.
3. Create a file named **m05report.txt** and paste in your debugger output:

#0 0x00000000004008b5 in funbottom ...

#1 0x00000000004008f2 in funtop ...

#2 0x0000000000400a8c in main ...

## Problem 5

Given the C function defined in the file **printproc.c**, write a Python program named **m05p05.py** that takes one parameter from the command line, calls the C function and passes it the proper parameters for a selected process.