# Spring 2014 - CS 4/510 - Topics in Software Testing

**Due: May 23 2014**

## Project 3: Program debugging and GDB

### Part 1: Debugging a Data Encryption Standard (DES) implementation

The Data Encryption Standard (DES) algorithm, adopted by the U.S. government in 1977, is a block cipher that transforms 64-bit data blocks under a 56-bit secret key, by means of permutation and substitution. It is officially described in FIPS PUB 46. The DES algorithm is widely used and is still considered reasonably secure.

In this part, we give an implementation of DES algorithm which includes at least 5 bugs, your task is to find and fix all bugs. In the DES folder, we give several files as follows:

**des.c**: buggy program
**bool.h, tables.h**: header files
**des.txt**: a step by step tutorial to implement DES
**des_correct**: an executable binary which can deliver correct results
**Makefile**: run “make” to compile the program

### Part 2: using GDB

GDB, the GNU Project debugger, allows you to see what is going on `inside' another program while it executes -- or what another program was doing at the moment it crashed.

In this part, we require everyone to master basic GDB commands to debug a program. Under the project folder, you can find a quick sort implementation which you can use for practice. On May 23, we will give another program and you need to show the following five functionalities:

- ❖ List all files, functions, global variables and locals
- ❖ Set line/function breakpoints
- ❖ List the path step by step triggered by the given test case
- ❖ Modify a specified variable at runtime
- ❖ Debug a running process

## References

- GDB
  - http://www.sourceware.org/gdb/
  - http://www.gnu.org/software/gdb/

# Spring 2014 - CS 4/510 - Topics in Software Testing

**Due: May 23 2014**

## Project 3: Grading sheet

| No. | Feature | % | Grade |
|---|---|---|---|
| | Given an program with bugs and a correct executable binary | | |
| 1 | Find 5 bugs in the program (5 points each) | 25 | |
| 2 | Fix 5 bugs in the program (5 points each) | 25 | |
| | | | |
| In-class | Use GDB to debug the given program | | |
| 3 | List all files, functions, global variables and locals | 10 | |
| 4 | Set line/function breakpoints | 10 | |
| 5 | List the path step by step triggered by the given test case, including th lines in the functions invoked | 10 | |
| 6 | Modify a specified variable at runtime | 10 | |
| 7 | Debug a running process | 10 | |
| | | | |
| Total | | 100 | |