# CS408: Design Inspection, Code Inspection and Unit Testing Grading Rubric

Do Design Inspection, Code Inspection, and Unit Testing on your software product. After each of the three steps, fix all defects found. Create a log (report) of all defects found by Inspection and Unit Testing and how they were corrected.

You should create a design for your product that shows major components. Start by inspecting that. List any defects found in the Design Inspection in the log. Then, move on to writing code. Inspect as many modules as you can. List any defects found in those Code Module Inspections in the log. Then, proceed to Unit Testing. List any defects found in Unit Testing in the log.

## 1 Design Inspection Defects (20 points)

The design inspection defects should reveal the defects which occur while combining the modules or during the interaction between the modules. We look at the how clearly you have understood the defects that could occur from a design perspective, the solutions/fixes provided and the severity assigned to each defect.

## 2 Code Inspection Defects (35 points)

For code inspection, we expect the defects arising from inefficient or wrong coding techniques which could result in exceptions, buffer overflow etc. We will also check for the solutions/fixes provided and the severity assigned to each defect.

Given below are two code snippets with a small change. One of them could result in a runtime exception. Identifying the defect would help you in your code inspection process.

(Hint: Think of buffer overflow)

```
 #include <string.h>
int main(int argc, char **argv) {
   char buffer [10];
   strcpy(buffer, ''test'');
}
```

```
 #include <string.h>
int main(int argc, char **argv) {
  char buffer [10];
  strcpy(buffer, argv[1]);
}
```

# 3　Unit Testing Defects (45 points)

- (10 points)
  Specify whether you automated the unit testing. If yes, give a brief description of your automation framework/strategy.

- (35 points)
  For unit testing, we expect you to clearly define modules of your product, to identify the input and output to each module, and to list down the defects found during the unit testing of each module. We will also check for the solutions/fixes provided and the severity assigned to each defect.

Refer to https://www.cs.purdue.edu/homes/suresh/408-Spring2017/defectlog.htm for formatting.