HOMEWORK ASSIGNMENT #1

CS589; Fall 2014
Due Date: **September 24, 2014**
Late homework 50% off
After **September 28** the homework assignment will not be accepted.

This is an **individual** assignment. **Identical or similiar** solutions will penalized. The **hardcopy** of the assignment must be submitted. Electronic submissions are not acceptable. Notice that the Blackboard homework assignment submissions are only considered as a proof of submission on time (before the deadline). If the hardcopy is different than the electronic version submitted on the Blackboard, then **50% penalty** will be applied.

SPECIFICATION-BASED TESTING

Suppose a software component (called a Grader component) has been implemented to automatically compute a grade in a course. A course taught at a university has two components: (1) two exams and (2) a project. To pass the course with grade C a student must score at least 50 points in the Exam-1, 60 points in Exam-2, and 50 points in the Project. Students pass the course with grade B if they score at least 60 points in the Exam-1, 65 points in Exam-2, and 60 points in the Project. If, in addition to this, the average of the exams and the Project is at least 75 points then students are awarded a grade A. Final grades for the course are: A, B, C, and E. The Grader component accepts six inputs:

Last name
First name
Student #
Exam-1
Exam-2
Project

Assumptions:

- Assume *Exam-1, Exam-2* and *Project* are integers.
- The ranges for the exam scores and the project score are:
  - $0 \leq$ *Exam-1* ≤ 100
  - $0 \leq$ *Exam-2* ≤ 100
  - $0 \leq$ *Project* ≤ 100
- The maximum size of the "*First name*" is 12 characters and "*Last name*" is 20 characters.
- *Student #* is a number represented as a 9-character string in the following format: AXXXXXXXX, where X is a digit.

Sample test cases for the Grader component:

Test #1: *Last name*=Smith, *First name*=John, *Student #*=A11112222, *Exam-1*=57, *Exam-2* = 64, *Project*=55

Test #2: *Last name*=Smith, *First name*=Mary, *Student #*=A42312242, *Exam-1*=75, *Exam-2* = 24, *Project*=85

PROBLEM #1 (35 points): Equivalence partition testing
Identify input conditions for the Grader component related to:

1. Last name
2. First name
3. Student #
4. Exam-1
5. Exam-2
6. Project

From the identified input conditions list equivalence valid and invalid sub-domains (classes). Based on the identified sub-domains design test cases using:

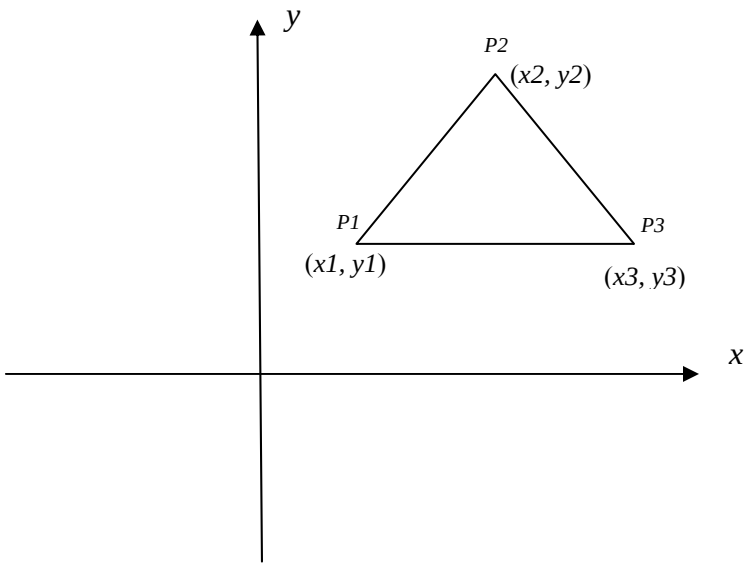
a. Strong normal equivalence testing,
b. Weak robust equivalence testing

Hint: Before designing test cases, identify related/unrelated input conditions.

PROBLEM #2 (30 points): Boundary-Value Testing
Based on the identified sub-domains in Problem #1 design:

1. Normal Boundary-Value Analysis test cases.
2. Robust Boundary Value test cases.

PROBLEM #3 (35 points): Decision-Table based testing
A *Triangle* program accepts six (6) integers: *x1*, *y1*, *x2*, *y2*, *x3* and *y3* as input. These are cordinates of 3 vertices of a triangle, where (*x1*, *y1*) represent coordinates of vertice *P1*, (*x2*, *y2*) represent coordinates of vertice *P2*, and (*x3*, *y3*) represent coordinates of vertice *P3*.



The output to the program is the type of triangle determined by three vertices:

- Equilateral triangle
- Isosceles triangle
- Scalene triangle
- Right triangle
- Not triangle
- Invalid input (for inputs that violate input conditions)

The following input conditions are identified for six inputs:

$$-100 \leq x1, y1, x2, y2, x3, y3 \leq 100$$

Use **decision-table based testing** to design test cases to test the *Triangle* program.
Provide a decision table and test cases derived from the decision table.

A sample test case for this program:
Test #1: *x1*=54, *y1*=72, *x2*=25, *y2*=72, *x3*=7, *y3*=17

Note: In your solution conditions in the decision table **cannot** be complex logical expressions, e.g.,

(x1=x2) and (y1=y2)

is **not acceptable** as a condition in the decision table.

The conditions in a decision table must simple conditions, e.g.,
(x1=x2) is an acceptable condition;
(y1=y2) is also an acceptable condition.