Software Testing Types – Q&A

1. Myers [Mye79] uses the following program as a self-assessment for your ability to specify adequate testing: A program reads three integer values. The three values are interpreted as representing the lengths of the sides of a triangle. The program prints a message that states whether the triangle is scalene, isosceles, or equilateral. Develop a set of test cases that you feel will adequately test this program.

2. Design and implement the program (with error handling where appropriate) specified in Question 1. Derive a flow graph for the program and apply basis path testing to develop test cases that will guarantee that all statements in the program have been tested. Execute the cases and show your results.

3. Give at least three examples in which black-box testing might give the impression that "everything's OK," while white-box tests might uncover an error. Give at least three examples in which white-box testing might give the impression that "everything's OK," while black-box tests might uncover an error.
   For specific input, an error occurs internally resulting in:
   1) Improper data placed in a global data area;
   2) Improper flags that will be tested in a subsequent series of tests;
   3) Improper hardware control that can only be uncovered during system test; yet "correct" output is produced.

4. Will exhaustive testing (even if it is possible for very small programs) guarantee that the program is 100 percent correct?

   No, even an exhaustive test (if it were possible) may be unable to uncover performance problems and errors in the specification of the software. In this case both input and output "equivalence classes" are considered. For each class, the student should identify boundaries based on numeric ranges, elements of a set, system commands, etc. This can be a paper exercise in which test cases for a GUI for some well know application are derived.

5. What are the attributes of a good software test?
   - Has a high probability of finding an error
   - Not redundant
   - Should be capable of uncovering a whole class of errors
   - Should not be too simple or too complex

6. Describe the differences between black-box testing and white-box testing.
   Black-box testing involves testing the functionality of a software component without knowing the details of its internal logic. White-box testing involves testing the independent logic paths with full implementation knowledge.

7. What is equivalence partitioning as it applies to software testing?

   A black-box testing technique in which the input domain is divided into classes of equivalent data items. Test cases are derived from combinations of elements from each equivalence class. Exhaustive testing of all input domain values is not necessary.

8. Describe three control structure testing strategies.

   Condition or branch testing -uses test cases that exercise every decision statement in the program.

   Data flow testing - selects test paths (definition use chains) according to the locations of variable definitions and uses in the program

   Loop testing -tests focus on the validity the repetition constructs (making sure that loops start and stop when they are supposed to)