Introduction to Software Analysis and Testing

In this course, you will work in teams of 3–4 students to learn new concepts. The first activity will introduce you to the process. Next we'll take a first look at software testing.

Meta Activity: Team Roles

Decide who will be what role for today; we will rotate the roles each week. If you have only three people, one should have two roles. If you have five people, two may share the same role.

Manager:	Helen Hu	keeps track of time, all voices are heard
Presenter:	Clif Kussmaul	asks questions, gives the team's answers
Recorder:	Chris Mayfield	quality control and consensus building
Reflector:	Aman Yadav	team dynamics, suggest improvements

Questions (15 min) Start time:

1. What is the difference between **bold** and *italics* on the role cards?

The bold points describe what the responsibilities are. Examples of what that person would say are in italics.

2. Manager: invite each person to explain their role to the team. Recorder: take notes of the discussion by writing down key phrases next to the table above.

3. What responsibilities do two or more roles have in common?

Both the presenter and the recorder help the team reach consensus. The manager and reflector both monitor how the team is working.

4. For each role, give an example of how someone observing your team would know that a person is not doing their job well.

- Manager: The team is constantly getting behind.
- Presenter: The student doesn't know what to say.
- Recorder: Some team members aren't taking good notes.
- Reflector: The student never comments on team dynamics.

Model 1 Triangle Program Testcases

Questions (15 min)

Start time:

5. Consider the following program and provide your testcases with explanations for why each testcase is needed:

```java
import java.util.*;
public class Triangle {
 public static void main(String[] args) {
  Scanner in = new Scanner(System.in);
  System.out.print("Input side1: ");
  int s1 = in .nextInt();
  System.out.print("Input side2: ");
  int s2 = in .nextInt();
  System.out.print("Input side3: ");
  int s3 = in .nextInt();

  if (!isValidTriangle(s1, s2, s3))
    System.out.println("These sides can't form a triangle");
  else if (isEquilateral(s1, s2, s3))
    System.out.println("These sides can form an equilateral triangle");
  else if (isIsosceles(s1, s2, s3))
    System.out.println("These sides can form an isosceles triangle");
  else
    System.out.println("These sides can form a scalene triangle");
 }
 public static boolean isValidTriangle(int a, int b, int c) {
  return (a + b > c && b + c > a && c + a > b);
 }
 public static boolean isEquilateral(int a, int b, int c) {
  return (a == b && b == c);
 }
 public static boolean isIsosceles(int a, int b, int c) {
  return (a == b || b == c || a == c);
 }
}
```

Answers may vary

Model 2 Triangle Program Testcases Checklist

Questions (15 min) Start time:

Specify which of your initial testcases falls under each category or provide an additional testcase otherwise:

6. Do you have a test case that represents a valid scalene triangle?

Answers may vary; 5,6,7
Note that test cases such as 1, 2, 3 and 2, 5, 10 do not warrant a yes answer because a triangle having these dimensions is not valid.

7. Do you have a test case that represents a valid equilateral triangle?

Answers may vary; 1,1,1

8. Do you have a test case that represents a valid isosceles triangle?

Answers may vary; 2,2,3
Note that a test case representing 2, 2, 4 would not count because it is not a valid triangle.

9. Do you have at least three test cases that represent valid isosceles triangles such that you have tried all three permutations of two equal sides?

Answers may vary; 3, 3, 4; 3, 4, 3; and 4, 3, 3

10. Do you have a test case in which one side has a zero value?

Answers may vary; 3, 3, 0

11. Do you have a test case in which one side has a negative value?

Answers may vary; 3, 3, -4

12. Do you have a test case with three integers greater than zero such that the sum of two of the numbers is equal to the third?

Answers may vary; 1, 2, 3

13. Do you have at least three test cases such that you have tried all three permutations where the length of one side is equal to the sum of the lengths of the other two sides?

Answers may vary; 1, 2, 3; 1, 3, 2; and 3, 1, 2

14. Do you have a test case with three integers greater than zero such that the sum of two of the numbers is less than the third?

Answers may vary; 1, 2, 4 or 12, 15, 30

15. Do you have at least three test cases such that you have tried all three permutations?

Answers may vary; 1, 2, 4; 1, 4, 2; and 4, 1, 2

16. Do you have a test case in which all sides are zero? Answers may vary

17. Do you have at least one test case specifying noninteger values?

Answers may vary; 2.5, 3.5, 5.5

18. Do you have at least one test case specifying the wrong number of values?

Answers may vary;
2, 2 -> the scanner object waits for int input

19. For each test case did you specify the expected output (or behavior) from the program in addition to the input values? Answers may vary

Model 3 Discussion

Questions (15 min) Start time:

20. The handout for the pre-class reading introduced you to two testing techniques: black-box and white-box. What are some similarities and differences between the two?

Both developers and testers can employ them, they both require specifying the input along with its expected output (or behavior). Black-box is focused on program specifications while white-box is focused on the program's logic, structure and internal behavior.

21. Which testing technique is superior (between black-box and white-box)?

Neither, they complement each other (although they may also overlap).

22. What kind of testing technique (black-box or white-box) did you employ in Model 1 and 2?

The checklist (Model 2) fits more the black-box testing because it's focused on the problem domain. White-box testing is focused on program logic and structure (could have been done in Model 1).

23. Are the testcases required by the checklist (Model 2) enough to properly test the triangle program? Did you come up with any additional testcases?

One could also try testcases with all negative integer inputs, or different testcases for each of the conditions inside each function. For example, one testcases in which `a == b` is true, but `b == c` and `a == c` are false, etc.

24. Are more testcases better than less testcases?

Not necessarily, in this course, we'll learn more established techniques of generating testcases and practice eliminating redundant testcases (economy of testing)

Model 4 Testing Principles

Questions (15 mins) Start time:

25. For the principle (or principles) assigned to your group provide the arguments (or examples) that the author makes in support of the principle.

Answers may vary

26. Do you agree/ disagree/ or both agree and disagree with the arguments or the examples? Also write down any interesting arguments, comments, or examples that come up in your group discussion.

Answers may vary

27. Reason and come up with examples of why the arguments provided by the author are enough (or not enough) to constitute a principle for testing.

Answers may vary