# FIT2107
# Week 2 Workshop Sheet

**Objectives**

The objectives of this workshop are:

- To understand the differences between fault, failure, and error.
- Practice the use of fault, failure, and error.
- To be able to apply quality attributes of software systems on a set of problems.

**Expectation**

**Participation** in these workshops will not be assessed towards final marks, but it should be considered **the absolute bare minimum** required effort to **pass** this unit. Nevertheless, participation is by no means the recommended level of effort if you wish to excel in this unit, your future units, and the workplace.

In a university setting, simply completing assigned work is not sufficient to maximise your learning outcomes, you will need to practice some self-learning as well. This is true in all units and we assist this by providing guidance on where to start your search with suggested readings and video links.

## Questions - to be tackled in groups of 3-4 students.

### Question 1: Software Quality Attributes (40 minutes)

A new system, MYDAM, has been introduced at a University very much like Monash, for lecturers and tutors to mark assignments. Unfortunately, there have been a lot of problems with the system. You hear a lecturer say:

*"This new MYDAM thingamy is a complete disaster! It takes over a minute to save a student's assignment marks after you hit the submit button – and that's if you can figure out the ridiculous marking interface – none of my tutors were able to use the system without help. Even when you enter the marks correctly, when you ask for total marks for calculating the final subject mark it rounds up rather than to the nearest integer, as university policy says we should. And, just yesterday, the server crashed and lost all the marks that were entered that morning!"*

There are multiple software quality problems described in this lecturer's comments. Identify the distinct quality problems and classify them in order of importance. Provide justification for your answer.

**Question 2: Fill in the Blanks (5 minutes)**

Having a certain terminology helps testers to explain the problems they have with a program or in their software

**Mark**: Hey, Jane, I just observed a (1) _ _ _ _ _ _ in our software: if the user has multiple surnames, our software doesn't allow them to sign in.
**Jane**: Oh, that's awful. Let me debug the code so that I can find the (2) _ _ _ _ _ _. *(a few minutes later)*
**Jane**: Mark, I found it! It was my (3) _ _ _ _ _ _. I programmed that part, but never thought of this case.
**Mark**: No worries, Jane! Thanks for fixing it!

**Question 3: Faults, Errors and Failures (40 minutes)**

Given a piece of code in Java programming language below, there is a simple fault in *numZero* method.

```
public static int numZero (int[] arr){
// Effects: If arr is null throw NullPointerException
// else return the number of occurrences of 0 in arr
    int count = 0;
    for (int i = 1; i < arr.length; i++)
        if (arr[i] == 0)
            count++;
    return count;
}
```

Answer using test cases i.e. test inputs

a) Where is the fault location in the source code? How the fault will lead to error and failure? Explain it.
b) How would you fix it?
c) Can the fault location be reached?

d) What is the output (Hint: your output is based on the test cases)?

Note: This is a good example to prepare you all for testing in a real sense which will begin from next week. Do not worry if your answers are wrong. Discuss your solutions with tutor.