

CS 3250: Software Testing (Fall 2026)

Activity: Fault, Error, Failure, and RIPR — solution

(no submission)

Purpose:

- Understand the differences between fault, error, and failure
- Understand RIPR model and use it to design tests
- Get ready to work on homework assignment; prepare for quizzes and the final exam

Consider the following methods

```
static void riprEx1 (int y)
{
    // int x = 0;           // correct code
    int x = 1;             // fault
    if (y >= x)
        System.out.println(y);
    else
        System.out.println(x);
}

static void riprEx2 (int y)
{
    // int x = 0;           // correct code
    int x = y;             // fault
    if (y >= x)
        System.out.println(y);
    else
        System.out.println(x);
}
```

Consider the following test inputs and identify whether the inputs cause the fault to be reached (and thus executed), infect the state of the method under test, and propagate and result in a failure.

Test inputs	Reach (execute) the fault	Infect the state (cause an error)	Propagate (result in a failure)
riprEx1(1)	X	X (x = 1, not 0)	(expect 1, actual 1)
riprEx1(0)	X	X (x = 1, not 0)	X (expect 0, actual 1)
riprEx2(1)	X	X (x = 1, not 0)	(expect 1, actual 1)
riprEx2(0)	X	(no error, x = 0 as it should be)	(expect 0, actual 0)

1. Consider the fault in `riprEx2()` method. If possible, find a test input that does **not** reach (i.e., not execute) the fault. If not, briefly explain why not. Make sure your test input is different from the given input values (table above).

Impossible. The fault is always reached as soon as the method is called

The reachability (R) condition is true.

2. Consider the fault in `ripEx2()` method. If possible, find a test input that **reaches the fault**, but does **not** result in an **error**. If not, briefly explain why not. Make sure your test input is different from the given input values (table above).

Input $y = 0$, expected output = 0, actual output = 0
The only possible input value that won't result in an error is $y = 0$

R: true
I: $y \neq 0$

3. Consider the fault in `ripEx2()` method. If possible, find a test input that results in an **error**, but **not a failure**. If not, briefly explain why not. Make sure your test input is different from the given input values (table above).

Input $y = 2$, expected output = 2, actual output = 2
Any $y > 0$ will cause an error but no failure.

R: true
I: $y \neq 0$
P: $y < 0$

4. Consider the fault in `ripEx2()` method. Find a test input that results in a failure. If not, briefly explain why not. Make sure your test input is different from the given input values (table above).

Input $y = -2$, expected output = 0, actual output = -2
To cause a failure, any $y < 0$ would work

R: true
I: $y \neq 0$
P: $y < 0$