

---

## Software Engineering

<http://proglang.informatik.uni-freiburg.de/teaching/swt/2009/>

---

### Exercise Sheet 11

#### Exercise 1: Simplifying Input (20 Points)

In this exercise, you are supposed to write a simple Java program that simplifies failure-inducing input. The idea is to use Delta Debugging to find a minimal input that causes an XML parser to fail. Download “parser” and “example.xml” from the website. The program “parser” has a small defect. The input file “example.xml” causes the parser to fail:

```
$ ./parser example.xml  
segmentation fault!
```

##### Exercise 1.1:

Write a `test` function. Start with a Java program that invokes the parser, as described above, and assesses the result. You may use the Java method `Runtime.getRuntime().exec()` for this. Take care to differentiate three outcomes:

1. The parser successfully parses the file (✓).
2. The parser fails as in the original failure (✗).
3. The parser has another outcome, in particular parse errors (?).

##### Exercise 1.2:

Write a `split` function. You may want to split the input along token delimiters.

##### Exercise 1.3:

Attach Delta Debugging by implementing the `ddmin` function from the slides.

##### Exercise 1.4:

Run your program. It should record all tests and the corresponding outcomes. How many tests did Delta Debugging take? What is the simplified failure-inducing input your program extracted from “example.xml”?

**Exercise 1.5:**

Make at least one suggestion how to

1. decrease the number of overall tests needed.
2. decrease the number of unresolved (?) test outcomes.