

## Static Test Techniques Exercise (10 Study points - non-mandatory)

### 1. Recap book chapter 1 – 2 (1 SP – 55% must be passed)

Do quiz on Moodle named “Intro to Testing (Black chap. 1-2)”.

### 2. Static Code Analysis of Triangle program (4 SP)

- Install Metrics software in your IDE (see tool examples in slides)
- Check coding standards in your Triangle program
- Calculate central metrics in your Triangle program – make screen dump
- Find out what CC variation that your metrics tool uses
- Possibly refactor your code based on static testing results. You might also want somebody else to review your code ☺ Write down what changes you make base on static techniques used.
- Write test cases in xUnit tool. Possibly refactor your code once (or twice more) in order to make the tests pass!

### 3. Peer Review Checklist (2 SP)

Smartbear has made a “[Best practices for Code Review](#)”. One of the things on their list is to use checklists in the review process (tip #6). Explain their [checklist](#) in your own words, make comments on what you especially find useful in the list and possibly provide with examples.

### 4. Review code that mysteriously fails its unit tests (1 SP)

Please review this code, as the tests do not succeed as expected. Can you fix it?

#### Code under test:

```
public class Catalog {
    private static List<Person> people = new ArrayList<>();

    public void addPerson(Person person) {
        if (Calendar.getInstance().get(Calendar.YEAR) - person.getYearOfBirth() < 18)
        {
            throw new IllegalArgumentException("Only adults admitted.");
        }
        people.add(person);
    }
    public int getNrOfPeople() {
        return people.size();
    }
}
```

### Test code

```
public class TestCatalog {  
  
    private final Catalog underTest = new Catalog();  
  
    @Test(expected = IllegalArgumentException.class)  
    public void addingAMinorShouldThrowException() {  
        assertEquals(0, underTest.getNrOfPeople());  
        Person p = new Person(2015);  
        underTest.addPerson(p);  
    }  
  
    @Test  
    public void addingAnAdultShouldSucceed() {  
        assertEquals(0, underTest.getNrOfPeople());  
        Person p = new Person(1985);  
        underTest.addPerson(p);  
        assertEquals(1, underTest.getNrOfPeople());  
    }  
}
```

### **5. Coding Standard Document (1 SP)**

List the coding standards - best practices and code conventions - that you find most important for a team to follow (½ - 1 page).

### **6. Highlights from lecture by Gitte Ottosen, Gapgemini-Sogeti (1 SP)**

Describe the three most essential things – in your opinion - that were mentioned in the guest lecture on February 12<sup>th</sup> (½-1 page description).

### **Formalities**

Hand-in on Moodle: Document with text descriptions + link to code on Github

Code Deliverables: Triangle Program (your best version ☺) + unit tests

Deadline: February 18th at noon