

## Tutorial 6

**Comment**

Tutorial exercises should be done without a computer

## 1 Substitution Part 1

	Parent Class	Child Class
1	$a < 0 \ \&\& \ b < 0$	$a \neq 0 \ \&\& \ b < 0$
2	$a \text{ instanceof } \text{Animal}$	$a \text{ instanceof } \text{String}$
3	$a \text{ instanceof } \text{Animal} \    \ b \text{ instanceof } \text{Zebra}$	$a \text{ instanceof } \text{Animal} \ \&\& \ b \text{ instanceof } \text{Zebra}$
4	$a \text{ instanceof } \text{Animal} \ \&\& \ b \text{ instanceof } \text{Zebra}$	$a \text{ instanceof } \text{Animal}$
5	$a \text{ instanceof } \text{Animal} \ \&\& \ b \text{ instanceof } \text{Object}$	$a \text{ instanceof } \text{Animal}$
6	$a \text{ instanceof } \text{Zebra}$	$a \text{ instanceof } \text{Tiger}$

For each of the above, could these pair of conditions be:

- Preconditions
- Postconditions
- Neither

## 2 Pre & Post Conditions

Write a Javadoc comment for the following method. Include `@ensure` and `@require` tags for all appropriate pre and post conditions.

```
public boolean q2(String[] strArray,  
int firstIndex, int secondIndex) {  
    return (strArray[firstIndex] == strArray[secondIndex]);  
}
```

As a challenge, consider how you would specify that the `q2` method does not change any elements in `strArray`.

### 3 Substitution Part 2

During lecture 5 you were introduced to the *substitution principle* as it applies to writing Java methods and classes. The following questions refer to the class structure given below.

```
class X {
    /**
     * @require fontSize > 5
     * @ensure \result >= 0
     */
    int detexify(Object symbol, float fontSize) {
        // Things happen here
    }
}

class Y extends X {
    /**
     * @require fontSize > 0
     * @ensure \result > 0
     */
    int detexify(Object symbol, float fontSize) {
        // Other things happen here
    }
}

class Z extends X {
    /**
     * @require fontSize > 5
     * @ensure \result > 0
     */
    int detexify(Object symbol, float fontSize) {
        // Do other things happen here?
    }
}
```

- Give one reason a programmer would choose to use a precondition.
- Does Y violate the substitution principle? Why/why not?
- How would you change the postcondition of class Z so that it violates the substitution principle?