

# CSE 5910 Software Foundations

## Sample Test 1

### 1 (20 marks)

In Java, `Strings` are *immutable*.

- (a) What does this mean?
- (b) What is an advantage of `Strings` being immutable?
- (c) When inspecting the API of a class, what may be an indication that the instances of the class are immutable?

### 2 (20 marks)

Consider the following API fragment.

```
public Student(String name, String number)
```

Creates a student with the given name and student number.

**Parameters:**

`name` - the name of the student.

`number` - the number of the student.

**Precondition:**

`number.length() == 9`.

The main method of an app contains the following statement.

```
Student me = new Student("Franck van Breugel", "123456789");
```

Assume this statement causes the app to crash. Who is responsible, the client or the implementer? Explain your answer.

### 3 (20 marks)

- (a) The `Fraction` class has the attributes `numerator` and `denominator`. Consider the following fragment of the main method.

```
Fraction f = new Fraction(1, 2);  
Fraction g = new Fraction(2, 4);  
Fraction h = new Fraction(2, 4);  
Fraction i = h;
```

Draw the corresponding memory diagram once the execution reaches the end of the above fragment. Make sure that the attributes `numerator` and `denominator` and the variables `f`, `g`, `h` and `i` are reflected in your diagram. Include class, object and invocation blocks.

- (b) The `Fraction` class has static attribute `isQuoted` (whose default value is `true`). Consider the following fragment of the `main` method.

```
final int NUMBER = 5;
boolean quoted = Fraction.isQuoted;
Fraction.isQuoted = false;
```

Draw the corresponding memory diagram once the execution reaches the end of the above fragment. Make sure that the attribute `isQuoted` and the variables `NUMBER` and `quoted` are reflected in your diagram. Include class, object and invocation blocks.

#### 4 (20 marks)

Consider the following fragment of the `main` method.

```
String u = new String("one");
String v = new String("one");
String w = null;
String x = "two";
String y = "two";
```

- (a) Object references can be compared using the `equals` method or the `==` operator. Fill each entry of the two tables below with either `true` or `false`.

<code>equals</code>	<code>u</code>	<code>v</code>	<code>w</code>	<code>x</code>	<code>y</code>
<code>u</code>					
<code>v</code>					
<code>w</code>					
<code>x</code>					
<code>y</code>					

==	u	v	w	x	y
u					
v					
w					
x					
y					

(b) Explain the differences in the two tables.

## 5 (20 marks)

Consider the following fragment of the `main` method.

```
long seed = 1234;
Fraction.setSeed(seed);
output.println(seed);
```

Can you say anything about the output that the above fragment produces? Explain your answer.