Lab sheet 2:
Inheritance in a Social Network Application

Setter: Idris Skloul Ibrahim

Attempt Task 2A and 2B, in the tutorial and finish 2C in the lab. Don't forget to commit your code before you leave the lab!

Deadline:

Task 2B Friday 11/3/2016 15:15 (end of lab)

Task 2C Friday 18/3/2016 17:15 (end of lab)

Task 2A: Have a look, and try to understand lab2.network1 and lab2.network2.

Use Dummie.java, to practice up and down casting.

(0 mark)

Name:……

Task 2B: Upcasting and Downcasting

Assume we have 4 classes:

1. **`Person`**
2. **`Teacher`**
3. **`Student`**
4. **`PhDStudent`**

`Teacher` and **`Student`** are both subclasses of **`Person`**. **`PhDStudent`** is a subclass of **`Student`**.

1. Draw a class inheritance tree. *(0.5 marks)*

myInheritanceTree

2. Which of the following assignments are legal, and why or why not? *(2.5 marks)*

a) `Person p1 = new Student();`

Explanation: _____

b) `Person p2 = new PhDStudent();`

Explanation: _____

c) `PhDStudent phd1 = new Student();`

Explanation: _____

d) `Teacher t1 = new Person();`

Explanation: _____

e) `Student s1 = new PhDStudent();`

Explanation: _____

3. Suppose we have the following legal declarations and assignments: *(3 marks)*

```
Person p1 = new Person();
Person p2 = new Student();
PhDStudent phd1 = new PhDStudent();
Teacher t1 = new Teacher();
Student s1 = new Student();
```

Based on these statements, which of the following assignments are legal and why or why not?

a) `s1 = (Student) p1;`

Explanation: _____

b) `s1 = p2;`

Explanation: _____

c) `p1 = s1;`

Explanation: _____

d) `t1 = s1;`

Explanation: _____

e) `s1 = phd1;`

Explanation: _____

f) `phd1 = (PhDStudent) s1;`

Explanation: _____

Task 2C: Introducing Deeper Hierarchies

(4marks)

1. Use the code in **`lab3.netwok3`**. In order to demonstrate that a subclass can access non-private elements of its superclass, try the following (slightly artificial) modification:

   - Create the method **`printShortSummary()`** in the **`MessagePost`** class, which should print out: "*Message post from NAME*", where NAME is the **`username`** from the superclass **`Post`**.

   - In order to access the (private) field **`username`** in **`Post`** you will have to create and call a **`getUserName()`** method.

2. Implement to new classes: **`EventPost`** and **`CommentedPost`**, and refactor the code to reflect the following inheritance hierarchy.
   Think: which methods should go where?