

## Tutorial problems – Wed 6 Nov and Fri 8 Nov 2019

**10min** Ask whether there are any questions. Collect these by writing them on the board and decide with the whole group which of these to address in the 10 minutes available for question answering. The answering of the question may also be put to the end of the tutorial.

**5min** Students to work in small groups (of 2 or 3) to answer:

**“Explain the idea of Inheritance.”**

Expected answer: Reuse and maintainability of code. Overriding only those methods that change.

**20min** The students should again work in small group of 2 or 3 to address the following problem with respect to Inheritance:

Declare a class **CreditCard** with fields **name**, **accountNumber**, and **amount** of appropriate types together with a constructor, and a setter and a getter for **amount**. (The full class would contain other methods, but these are not of concern here.)

A bank wants to offer different types of credit cards, also a **GoldCard** which has the additional field **fee**. Write a class **GoldCard** with a constructor, in which the **fee** is subtracted from the **amount** on the creation of objects. Make use of inheritance to reuse the code from above. (The full class would contain methods, which are not of concern here.)

**15min** The students should again work in small group of 2 or 3 to address the following problem with respect to Inheritance:

Assume the interface **I** and the three classes **A**, **B**, and **Test** be given as follows:

```
public interface I {
    void f();
}
public class A implements I {
    public A() {
    }
    public void f() {
    }
}
public class B extends A {
    public B() {
    }
    public void g() {
        System.out.print("g applied");
    }
}
public class Test {
    public static void main(String[] args) {
        I i = new I();
        A a = new A();
        B b = new B();
        i.f();
        a.f();
        b.f();
        i.g();
        a.g();
        b.g();
        B a1 = a;
        A b1 = b;
    }
}
```

For every line in the **main** method, give a reason whether the line will be accepted by the compiler or not.