# Assignment 4

**NOTE: exercises with prefix *** are not mandatory**

**Topics: Inheritance, Polymorphism**

1. Please read the following tutorial:
   *http://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html*

2. Given the class *Student* of assignment 2, extend it to create a specialized class *TAStudent* of teaching assistants. A teaching assistant is a student which was assigned some support tasks in one or more classes.

3. Now create a class *PhDStudent* to describe PhD students. A PhD student is a student with a master degree striving for a higher degree. They may be assigned teaching support as well.

4. Implement the methods *clone()*, *equal()*, and *toString()* for all types of students. Test the classes in 2 and 3 in a *TestStudent* class.

5. Given the following variables:

   ```
   TAStudent     ta;
   PhDStudent    phd;
   Student       student;
   ```

   And given the following *display()* methods:

   ```
   In TAStudent      void display() { System.out.println("I am a TA"); }
   In PhDStudent     void display() { System.out.println("I am a PhD student"); }
   In Student        void display() { System.out.println("I am a regular student"); }
   ```

   Explain what happens with the following instructions (assume the corresponding objects are already instantiate; take each case independently from each other):

   Case 1)
   ```
   student  = phd;
   student.display();
   ```

   Case 2)
   ```
   student  = (Student) phd;
   student.display();
   ```

   Case 3)
   ```
   ta = student;
   ta.display();
   ```

   Case 4)
   ```
   ta = (TAStudent) student;
   ta.display();
   ```

   Case 5)
   ```
   ta = phd;        // here it will depend on your own sub-classification
   ```

*ta.display();*

6. Given the variables of point 5, consider:

   ```
   student = phd;       // line 1
   student.display();   // line 2
   ```

   Which *display()* method is run: the one in the base class or the one in the derived class? Modify, if necessary, line 1 in order to get the call to *display()* in line 2 run the method in the base class.
   Whatever modification you perform, be it a casting, another method call, etc., the variable *student* MUST appear on the left side of the statement and the variable *phd* MUST appear on the right side of the same statement. You are not allowed to modify the displayI() methods. You are not allowed to add new custom methods either.

7. ***Create an array that contains 6 students: 2 regular students, 2 PhDs and 2 TAs. Implement a function that randomly picks an integer number between 0 and 5 used to index the corresponding element in the array. This element has to be printed out.

8. Imagine to extend class *Student* with a dummy class *DummyStudent*

   ```
   public class DummyStudent extends Student {

      private String completeName = "Mr.Some Name";

      public DummyStudent () {
         super(completeName);      // Call the base constructor
      }

   }
   ```

   Is there any problem? Comment and explain.

9. ***Explain what overriding and what overloading means and highlight the differences between these two concepts.