 Registration Number: _____

In-class Test 2

MSc/ICY Software Workshop

Assessed In-class Test: 15% of this term's continuous assessment mark.

Submission: Friday, 5 December 2014, 9:50 hours
No late submission

Usual examination conditions apply. You may not use any material during this in-class test.

Exercise 1: (Basic, 40%) Let a class `Exercise` be given as follows:

```
public class Exercise {

    private String difficulty;
    private String text;

    public Exercise(String difficulty, String text) {
        this.difficulty = difficulty;
        this.text = text;
    }
////////////////////////////////////[getters omitted]
    public String toString() {
        return String.format("Exercise (%s):\n%s",
                             getDifficulty(), getText());
    }
}
```

Write a subclass `AssessedExercise` of this class which has the additional field variable `maximalMarks`. Write the field variable(s), the constructor and the `toString()` method. Make use of inheritance as far as possible.

Exercise 2: (Medium, 30%) Let a list of type `int` be given. Write a **recursive** static method `public static int min(List list)` that returns the smallest element in the list. Note that if the list is empty, the method should throw an `IllegalStateException`.

Exercise 3: (Advanced, 30%) Write under each of the three classes what Java will do:

```
public class A {
  private int weight;
  private String unit;

  public A(int weight,
           String unit) {
      this.weight = weight;
      this.unit = unit;
  }

  public int getWeight() {
      return weight;
  }

  public String getUnit() {
      return unit;
  }

  public String toString() {
      return this.weight +
             getUnit();
  }

  public static void
        main(String[] args) {
      A a = new A(5, " kg ");
      System.out.println(a);
  }
}
```

```
public class B extends A {

  public int size;
  public String unit;

  public B(int weight,
         String wUnit,
         int size, String unit){
      super(weight, wUnit);
      this.size = size;
      this.unit = unit;
  }

  private int getSize() {
      return size;
  }

  public String getUnit() {
      return unit;
  }

  public String toString() {
      return super.toString() +
          getSize() + getUnit();
  }

  public static void
         main(String[] args) {
      B b = new B(4, " kg ",
                  5, " cbm ");
      System.out.println(b);
  }
}
```

```
public class C extends B {

  public String colour;

  public C(int weight,
      String wUnit,
      int size,
      String sUnit,
      String colour) {
    super(weight, wUnit,
          size, sUnit);
    this.colour = colour;
  }

  public String toString() {
      return colour + " "
             + getSize() ;
  }

  public static void
     main(String[] args) {
      C c = new C(3, " kg ",
         4, " cbm ", " blue");
      System.out.println(c);
  }
}
```