Lab7: Objects First

Answer each question in a different directory, named Lab7-[number]. Make sure that you include all the .java files we need in order to run your code. Unless otherwise specified the class that contains the main method should be named Lab7[number]Test We should be able to compile and run your code on LCPU (e.g. by typing `javac *.java` **and** `java Lab7[number]Test`**).**

All exercise directories should have ¡username¿-Lab7 as their parent directory. This folder should be compressed into a single file prior to submission. File submissions to moodle must be a zip or tar archive, the file name should be of this form:
`<username>- lab<number>.tar.gz` **or** `.zip`
For example: `ddk20-Lab7.zip` **or** `ddk20-Lab7.tar.gz`

There are often various ways of solving an exercise and some parts are left intentially vague. In those cases, you can decide how you code the exercise.

If you have questions about this lab sheet or are stuck with one of the questions talk to the tutors in the lab or post your question on Moodle. If your question needs to include code, please email it to the mailing list rather than posting it to Moodle.

This lab is part of your lab coursework. **The deadline for submission is 10am Tuesday 27 November 2012**

The code associated with the this lab sheet (Picture, TicketMachine and Book-exercise project.) can be downloaded from Moodle.

1 Adapting Existing Code

Exercises 1. In the source code of the class Picture (Picture Project), find the part that actually draws the picture. Change it so that the sun will be blue rather than yellow. Demonstrate the behaviour in the tester class.

5 marks

Exercises 2. In the Picture project, make a second sun appear in the picture. Demonstrate the behaviour in the tester class.

10 marks

Exercises 3. Add a sunset to the single-sun version of picture project by adding a new method. Demonstrate the behaviour in the tester class. Hint: you may want to have look at the different methods each class offers.

15 marks

Exercises 4. Use the TicketMachine project. Implement a method called empty that simulates the effect of removing all money from the machine. This method should have a void return type, and its body should simply set the total field to zero. Does this method need to take any parameters? Test your method by creating a machine, inserting some money, printing some tickets, checking the total and then emptying the machine.

10 marks

2 Writing Classes

Exercises 5. Create a Java class for triangles, named Triangle. Provide fields (with data type float) for the three side lengths of the triangle (with default value 1 for each side), accessor and mutator methods for all fields, methods to compute the area and the perimeter, and two constructors: one without parameters and one with the three side lengths as parameters. The setter methods shall print an error message in case an argument is lower than or equal to zero. Add a method:

```
public static void main(String[] args)
```

which creates an object of class Triangle with side lengths 10, 11.3, 13.1 and prints the area and the perimeter of this object.

15 marks

Exercises 6. Create a Java class for bank accounts, named BankAccount (like in Lab 5). Provide appropriate fields and methods. Make sure your code is robust. Provide a tester class to demonstrate that your code is working correctly.

35 marks

3 Optional Exercises

Exercises 7. Use the Book-exercise project. Add two accessor methods to the Book class - getAuthor and getTitle - that return the author and title fields in their respective results. Write an appropriate test class for your Book class.

Exercises 8. Use the Book-exercise project from the previous exercise. Add a further field, refNumber, to the Book class. This field can store a reference number for a library. It should be of type String and initialised to the zero length string ("") in the constructor as its initial value is not passed in a parameter to the constructor. Instead, define a mutator for it with the following signature:

```
public void setRefNumber(String ref)
```

The body of this method should assign the value of the parameter to the refNumber field. Add a corresponding getRefNumber accessor to help you check that the mutator works correctly. Demonstrate your code in the tester class.