Linnæus University
School of Computer Science,
Physics and Mathematics

Examination

Examiner Dr Jonas Lundberg	*The student complete the form below*
Date 2013–03–19 **Time** 9–14	*Name:* ______
Place Room D2236, D-building	*Address:* ______
Course Code 1DV007	______
Allowed aids None.	*E-mail:* ______
	Telephone: ______
	Civic reg. number: ______
Messages from the teacher Exercises: 5 Maximum points: 50 p Pass: 25 p	*The number of sheets handed in:* ______

Tick the exercises you hand in

Exercises	**Points**	*Put a tick*
1		
2		
3		
4		
5		

Points	**Grade**

Uppvisat kårlegitimation	☐ Ja	☐ Nej
Uppvisat legitimation	☐ Ja	☐ Nej

Tid för inlämmnande	**Tentamensvaktens signatur**

Linnæus University
School of Computer Science,
Physics and Mathematics
Dr Jonas Lundberg

Examination in Computer Science, 1DV007, 7.5cr

March 19, 2013, 9.00–14.00

Allowed aids: None.

1. **(a)** In the *Fibonacci* sequence the first two numbers are 0 and 1 and the others are the sum of the two previous numbers.

```
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...
```

Write a recursive Java method `int fib(int n)` that computes the n:th number in the Fibonacci sequence. (3p)

(b) Why is the above recursive method bad if you would like compute the first 50 numbers in the fibonacci sequence? Motivate your answer using an example. Also, present Java code for a much better *non-recursive* approach to compute and print the the first 50 numbers in the fibonacci sequence. (3p)

(c) Write a recursive method `int mult(int a, int b)` that computes the multiplication $a \cdot b$ with the use of addition. You can assume that both a and b are positive. (4p)

2. **(a)** What is an *algorithm*? What properties do we expect from an algorithm? Why do we use algorithms? (5p)

(b) Write an algorithm in pseudo code that is searching for an integer N in a sorted list (lowest first) using the method *binary search.* (5p)

3. **(a)** *Hashing* is an implementation technique often used when implementing certain data structures. Describe how hashing works and why it is used. (5p)

(b) The Java interface `java.util.Map` describes a *map* (or *table*) data structure. What is a map and what operations do we associate with a map? Show with a Java example how maps can be used in Java. (5p)

4. **(a)** What is *binding* in object-oriented programming languages. What is the difference between *static* and *dynamic* binding? How does binding work in Java? (5p)

(b) What separates an *abstract class* from an *interface* and an *ordinary* class in Java? (5p)

5. Consider the Java source code fragment below representing a binary search tree for integers where the methods `add` and `contains` are incomplete. Write down Java code for what these two methods should look like. Notice that the binary search tree should work as an integer set. That is, it should not be possible to store two elements representing the same integer value. (10p)

```
public class IntBST {
   private BST root = null;

   public void add(int n) {
      if (root==null)
         root = new BST(n);
      else
         root.add(n);
   }

   public boolean contains(int n) {
      if (root==null)
         return false;
      else
         return root.contains(n);
   }

   private class BST {
      int value;
      BST left = null;
      BST right = null;

      BST(int val) { value = val; }

      void add(int n) {          // Provide implementation for
         ...                     // this method.
      }

      boolean contains(int n) {  // Provide implementation for
         ...                     // this method.
      }
   }
}
```

Notice! You only have to hand in Java code for `BST.add(int n)` and `BST.contains(int n)`.

Good Luck!