ENGLISH

UNIVERSITY OF OSLO

Faculty of mathematical and natural sciences

Exam in :	INF1000 — Introductory course in object-oriented programming
Day of exam :	Thursday, December 2, 2004
Time :	09.00 – 12.00
Number of pages:	11 pages
Appendices :	None
Allowed to bring with you :	All written and printed material

- Read through the entire examination set and check that your examination set contains all 11 pages before you answer any questions. If you miss any information, you may make your own assumptions as long as they are in accordance with the spirit of the exam. Such assumptions must be explained.
- You write your answers directly into these pages, not on separate sheets of paper. This applies to all questions, both multiple choice questions and questions where you are asked to write program code. In the questions where you are asked to write program code, we recommend that you first make a draft on a separate sheet of paper and then write your answer on this examination form at the appropriate place.
- Some of the questions are multiple choice questions. For these questions your score depends on how many correct answers you give. No score is given if you leave a question unanswered, or if you mark both answer alternatives.
- If you have marked an alternative with a cross and you later determine that you do not want a cross there, you can write "FEIL" immediately to the left of the box.
- Except in question 6, you do not need to add comments to your program code.

Question 1

How many double values can be stored in each of these arrays?

a) **`double[][] tableA = new double[9][9];`**

Answer: ....

b) **`double[][] tableB = new double[10][2];`**

Answer: ....

c) **`double[][] tableC = new double[5][1];`**

Answer: ....

Question 2

Are these program sentences legal in Java?

YES	NO	
□	□	`int[] a = null;`
□	□	`int x = new int[50];`
□	□	`int[] x = int[50];`
□	□	`int[] verdier = new verdier[50];`
□	□	`int[] verdier = new int;`
□	□	`double[] x = new int[5];`
□	□	`String[] s = {"January", "February", "March"};`
□	□	`double[] x = new double{1.0, 2.0, 3.0, 4.0, 5.0};`
□	□	`double[] x, []y;`

Question 3

Suppose the following lines of program code are performed:

```
int a = 33;
int b = 65;
a = a + b;
b = a + b;
boolean equal = (a == b);
```

What is the value of the variable **`equal`** right after this?

Answer: ....

Question 4

Write the method below that calculates and returns price without tax, given the price with tax. Assume that the tax rate is 24%.
Answer:

```
double priceWithoutTax(double priceWithTax) {











}
```

Question 5

Suppose the following array has been declared in a method in a program:

```
int[] a = new int[100];
```

Suppose furthermore that the array **a** contains various values and that the value 0 occurs exactly once in the array. Are the alternatives below always going to give the result that **a[k]** contains the value 0 when the loop has completed?

```
YES     NO
 □       □      int k = 0;
                while (k < a.length && a[k] == 0) {
                   k++;
                }

 □       □      int k = 0;
                while (k < a.length && a[k] != 0) {
                   k++;
                }

 □       □      int k = 0;
                do {
                   k++;
                } while (a[k] != 0);

 □       □      boolean found = false;
                int k = 0;
                while (k < a.length && !found) {
                   if (a[k] == 0) {
                      found = true;
                   }
                }

 □       □      int k = 0;
                for (int i = 0; i < a.length; i++) {
                   if (a[k] == 0) {
                      k = i;
                   } else {
                      k++;
                   }
                }
```

Question 6

Suppose the following program lines are performed:

```
int sum = 9;
int value = sum++;
sum = ++sum + value++;
value += sum * 10;
```

What value does the variable **value** have right after the above sentences has been performed?

Answer: ....

Question 7

Here you should write **javadoc** comments as well as program code.

a) Finish writing the method below that is to return with the average of the values in the array **data** (i.e. the sum of all elements in the array **data** divided by the number of such elements). You should assume that the length of the array is at least 1.

b) Write a short **javadoc** comment to the method to tell what it does (i.e. what value it returns) and what the parameter is.

Answer:

```
double average(double[] data) {

}
```

Question 8

Given two variables

```
int i1, i2;
```

that are initialized to separate values. Write some program sentences that together switch the contents of the variables **i1** og **i2**. You may declare extra variables if you want to.

Answer:

Question 9

Finish writing the method below that is supposed to rearrange the elements in the array **x** such that the elements have the opposite order. Thus, if for example the values in **x** are in increasing order at the outset (**x[0] <= x[1] <= .. <= x[999]),** then the method arranges them in decreasing order. You should not declare any extra arrays to solve this problem, but you may declare some simple variables if you want to.

Answer:

```
void reverse (double[] x) {

}
```

Question 10 (Difficult)

Finish writing the method below that is supposed to print to screen the largest two values in the array **`v.`** A value may occur several places in the array, but not all values in the array are equal. If, for example, the array contains the value 100 two times, the value 50 two times, and all the remaining values are 0, then the method should print to screen that the largest value was 100 and the next to largest value was 50. You do not have to print to screen the location of these values in the array. You may assume that the length of the array is at least 2 and that it contains only non-negative integers.

```
void findTwoLargest (int[] v) {

}
```

Question 11

In the program below, write a constructor to class Circle that has the radius of the circle as its parameter. Also write an object method in class Circle that calculates and returns the area (you may use the formula: area = 3.14 * radius * radius). In addition, write program code in main to create two circles, one with radius 10.4 and one with radius 244.99. From main you then call the method for computing the area of the circle in each of the two Circle objects and print to screen one line for each circle, each line giving the radius and area of the circle.

```
class Circle {
     double radius;

    // write constructor here



   // write object method here to calculate and return the area



} // end class Circle


class CircleTest {

    public static void main (String[] args) {
         // write program code here that makes two Circle-objects
         // and prints out to screen their radius and area as
         // described above



   } // end main

} // end class SirkelTest
```

Question 12

In an elementary school there are 7 grades. Each grade consists of one or more groups. Each group consists of 10-20 pupils. Draw a UML class diagram with the four (Java) classes that may be used to represent this problem. Name the classes in the diagram. The diagram should include numbers on all relationships.

Answer:

Question 13

Suppose the following program is run:

```
class Tree {
  Tree left, right;
  static int count = 0;

  Tree (int x) {
    count++;
    if (x > 10 ) left = new Tree(x-10);
    if (x > 20 ) right = new Tree(x-20);
  }

  public static void main(String[] args) {
    count = 0;
    Tree b1 = new Tree(11);
    System.out.println(”1) ” + count);

    count = 0;
    Tree b2 = new Tree(21);
    System.out.println(”2) ” + count);

    count = 0;
    Tree b3 = new Tree(31);
    System.out.println(”3) ” + count);
  }
}
```

First make a drawing for yourself to find out how many Tree objects are made in each of the sentences in the main method. Then answer the following question: what is the program printing on the screen?

Answer: ....

Question 14

Suppose you have a text file **`Book.txt`** that contains, e.g., the text of a novel. Write a complete, compilable program that reads the file word by word and counts the number of occurrences of each word in the file. Your program should then create a new file **`WordOccurrences.txt`** and write the results of the analysis to this file. The following pieces of information about **`Book.txt`** is to be written to the file: (a) the total number of words; (b) the number of distinct words; and (c) a list of all words that occur at least 6 times. Each line of this list should consist of one word and the number of occurrences of that word, separated by blanks.

You may assume that the file **`Book.txt`** contains letters and blanks only, i.e. special characters like comma, period, colon etc do not occur in the text. You may also assume that all words are written in lower case, so you don't have to deal with problems caused by capital letters in the beginning of a sentence etc.

Hint: use a HashMap to keep account of the words you read in **`Book.txt`** and the number of occurrences of them. Think carefully through what to put into the HashMap. Recall that both the key and the value that you put into a HashMap must be (pointers to) objects. You may want to define a class Word where each object stores a word and the number of occurrences of that word.

Answer:

(continues on next page)

(continues on next page)