

Techniques of Java Programming

ETH Zurich

Date: 11 July 2006

Family name, first name:

Student number:

I confirm with my signature, that I was able to take this exam under regular circumstances and that I have read and understood the directions below.

Signature:

Directions:

- Exam duration: 90 minutes.
- Use a pen (**not** a pencil)!
- Please write your student number onto **each** sheet.
- All solutions can be written directly onto the exam sheets. If you need more space for your solution ask the supervisors for a sheet of official paper. You are **not** allowed to use other paper.
- You must answer all questions (no questions are optional).
- All personal documents are authorized. Exchanging documents during the examination would mean failing the examination.
- Only one solution can be handed in per question. Invalid solutions need to be crossed out clearly.
- Please write legibly! We will only correct solutions that we can read.
- Manage your time carefully (take into account the number of points for each question).
- Please **immediately** tell the supervisors of the exam if you feel disturbed during the exam.

Good luck!

Question	Number of possible points	Points
1	16	
2	12	
3	30	
4	13	
5	6	

1 Middleware and Publish Subscribe (16 Points)

Explain the role of stubs and the role of skeletons in middleware systems (4 points) :

Stub:

.....

.....

.....

.....

.....

Skeleton:

.....

.....

.....

.....

.....

Explain what marshalling and what unmarshalling is: (6 points)

.....

.....

.....

.....

.....

Give an example:

.....

.....

.....

.....

.....

Compare the traditional middleware infrastructures provided by frameworks like Java's RMI mechanism or CORBA with T-Spaces. Give example uses of each where you see fit: (6 points)

.....

.....

.....

.....

.....

Example use of Java RMI or CORBA:

.....

.....

.....

.....

.....

Example use of T-Spaces:

.....

.....

.....

.....

.....

2 Generics and Autoboxing (12 Points)

Read through codes in Figure 1

```
List<Integer> ins = Arrays.asList(1, 2, 3);  
int s = 0;  
for (int n: ints) {s += n;}  
assert s==6;
```

Figure 1: Generics and Autoboxing Example

In above source code, can I replace `List<Integer>` with `List<int>`? If not, why? (2 point)

```
List ins = Arrays.asList(---1---);
int s = 0;
for (---2---) {
    int n = (---3---);
    s += n;
}
assert s==6;
```

Figure 2: Generics Implementation

.....
.....

Explain two generic implementation techniques. (4 points)

.....
.....
.....
.....
.....
.....
.....

Please fill in the blanks in Figure 2 with the correct Java 1.4 instructions (no autoboxing, no generics). (6 Points)

- (1).....
- (2).....
- (3).....

3 Byte code (30 Points)

Question 1: Translate back to Java the following method (10 points):

.....
.....
.....
.....

```
public static synchronized int inc1();
```

Code:

```
0:  iconst_0
1:  istore_1
2:  goto    18
5:  iconst_1
6:  istore_0
7:  getstatic    #10; //Field a:I
10: iload_0
11: iadd
12: putstatic    #10; //Field a:I
15: iinc    1, 1
18: iload_1
19: bipush  10
21: if_icmplt    5
24: getstatic    #10; //Field a:I
27: ireturn
```

.....
.....
.....
.....

Question 2: How can one optimize the method previously shown? (8 Points)
Write the resulting byte code (7 points).

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

.....

.....

.....

.....

Question 3: How one could make such an optimization automatically (5 points)?

.....

.....

.....

.....

4 Non Blocking Input/Output (13 Points)

What is the difference between *Blocking Input/Output* and *Non-blocking Input/Output* (5 Points)?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

What are the advantages of *Blocking Input/Output* (2 Points)?

.....

.....

.....

.....
.....
.....
.....
.....
.....

What are the advantages of *Non-blocking Input/Output* (2 Points)?

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

In a program, why would you use *Non-blocking Input/Output* if you can use multiple threads of execution (4 Points)?

.....
.....
.....
.....
.....
.....
.....
.....
.....

.....

5 General Question (6 Points)

What are the advantages of using abstract classes over interfaces when developing a platform? Explain your point of view.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....