Programming Languages and Paradigms

Isak Karlsson, Tobias Fasth and Beatrice Åkerblom
(isak-kar | tobi-fas | beatrice)@dsv.su.se

February 27, 2012

Instructions

This exam is probably a lot of work. One reason is that the work is "unbounded" – there is no right answers and you can always go over your answer again. Knowing when to stop and what is "good enough" is an important quality. It is easy, but wrong, to think that quality is related to the time you spend on your answer.

You should use the literature, but you *may not copy* any answer, not discuss these questions with anyone else, or allow anyone to aid you in the answering of these questions. *Your answers must be products of your own mind.*

NOTE: The sum of your answers must not exceed 10 pages (LaTeX, article, 10pt) or 3500 words (whichever you prefer), *excluding* the bibliography. Divide this between your answers as you please. Don't exceed this limit!

There are no deliberate trick questions in this exam. However, in order to keep the questions reasonably short, we will assume the following:

1. We assume that you will read the questions with a critical mind. This means that there might be more to the question than what is explicitly stated. For example, if the question states "argue why X is better than Y", it might be the case that X is sometimes better than Y and sometimes the worse, depending on Z, or something else.
2. We assume that you understand that some of the questions have no single correct answer. A question can seldom be answered by a single statement. Rather, the questions are meant to be "thought provoking" – we hope that your knowledge and programming language skills will be visible in the *reasoning* in your answers.
3. We assume that you are able to determine if a question is very broad and make appropriate delimitations to be able to produce a sensible answer.

How to pass this exam

If you try to give the *correct answer* for each question, this exam is incredibly hard. Instead, aim for an answer that is *good enough*. While it may be hard

to determine what good enough is, it is much easier than covering all possible aspects. In your answers, we want *you* to think and to be able to make reasonable judgments and assumptions[1] – not give an answer that cannot possibly be questioned.

Core-dumping is not acceptable. A long answer with a lot of irrelevant remarks is worse than one that is short and to-the-point. Please note that we will also judge your ability to keep irrelevant information out of the answers.

A good strategy is to complete the exam the first or second day of the exam period, let it rest for an entire day, and then go back to polish your arguments. As you will most likely notice, this is a good strategy for any text you produce.

Be sure to properly reference all your sources. Also, keep a critical mind when searching for material.

Grading Every question is graded with the full grade scale. The final score is the average. The examiner may round up or down depending on the overall quality of the exam. For example, a strong E/D/B/C/A will count favourably in a "rounding situation".

To get a "pass" on a question, an absolute minimum requirement is that you answer all parts of the question. To score high, you should show that your view is well-informed and based on sound reasoning.

Handing In Hand in the electronic version of the exam in to Turnitin no later than 2012-01-09, 12:55.

First, go to `www.turnitin.com`

Create a user profile. You'll need the class ID and enrollment password to enroll in the class.

- **class ID:** 4657774
- **enrollment password:** iGNUcius

When you have created your profile, you just have to click on "PROP - Exam" and follow the instructions for submission of papers. Submit to the folder "Exam". You can choose file upload or cut & paste. Click "training" if you find the instructions unclear.

If you hand in late, expect it to negatively influence your score. The handed in file should be a pdf file, contain your name and be named after you, e.g. if Tobias were to hand the exam in, he would submit a file called *tobiasFasth.pdf*.

Hand in a paper version of your exam at the seminar 8, 2012-01-09, 13.00.

Good luck!!

Beatrice, Isak & Tobias

[1] By this, we implicitly say that answers produced by quoting extensively from different sources without criticizing, comparing, judging, etc. are not acceptable.

Questions

1. Parsing? **(30%)**

Carefully explain how the parsing process works (lexical analysis), all the different stages it consists of and their respective products. What choices can be made when implementing a parser? Give examples of where choices can be made, what the alternatives are and explain their pros and cons. Can parsers be useful in other situations than for compiler construction? Give examples.

2. Adding Logic? **(15%)**

Programming language designers have proposed that logic programming languages should be integrated with other programming paradigms such as functional programming. Argument and discuss both in favour for and against such integration.

3. Functional? **(15%)**

Choose *five* features of the functional paradigm that you think are important. Describe what they are and discuss why you think they are important. Within your discussion, give examples of them, either with code pseudocode or a diagram. Use examples from other paradigms as contrasts if needed.

4. Class-based vs. Prototype-based **(15%)**

Explain in detail the concepts inheritance and delegation used in object-orientation. Use examples to illustrate your description. Is there really any difference between them or does the use of them have equal power? Motivate your answer and use examples to illustrate.

5. Choosing Between Languages? **(25%)**

A supermarket chain, Maxi-Mart, wishes to make its product database available for customers over the web, initially for browsing and making shopping lists, but eventually also for online shopping. Maxi-Mart currently uses C++ for all its IT systems but is considering moving to another language (at least) for the web system. As the senior programmer, you have been given the task to write a report on evaluating the possible choices of language. Choose languages from two different programming paradigms, e.g. object-oriented, logic, functional programming, or imperative, and evaluate their strengths and weaknesses for such a web-based system. Motivate your choice of language, explain what paradigm the language belongs to and why. Also discuss and evaluate their strengths and weaknesses for the development of the new system.