Dynamic Programming Languages

Isak Karlsson and Beatrice Åkerblom
(isak-kar | beatrice)@dsv.su.se

March 13, 2012

Instructions

Bonus points already earned can be used to "buy" questions. A bought question will be counted as full score when the total is calculated. Note which questions you buy on the first page of the exam.

For each seminar you attended, you have earned 15% of the total exam. In all this means that you can have up to 45% worth of "exam currency" and to pass the exam, you will still need to answer at least two questions with at least the grade E.

Any bonus points that you don't use are forfeit.

Ideally, each answer should be around a page and a half, except question number 3 which probably should end up a bit longer. Shorter is better than longer, given that the same amount of information is contained.

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 help you in the answering of these questions. *Your answers must be products of your own mind.*

Grading Every question is graded with the full grade scale (which may be different depending on who you are and when you registered). The final score is the average. The examiner rounds 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 2012-03-17, 23.59.

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:** 4956063
- **enrollment password:** guido

When you have created your profile, you just have to click on "DYPL_12" and follow the instructions for submission of papers. Submit to the folder "Take-home 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 Isak were to hand the exam in, he would submit a file called *isakKarlsson.pdf*.

Questions

1. Python vs. Ruby **(15%)**
What are the major differences in terms of design philosophy between Ruby and Python? Are there any similarities? In what ways are these design philosophies visible in the languages? Give examples, discuss, explain them and motivate your choices.

2. Reflection **(15%)**
Explain the term reflection, how it can be useful in software development and if there are any hidden pitfalls. Also provide one well explained and motivated example.

3. Typing? **(40%)**
Ideally static typing guarantees that type errors do not occur at run-time. But what does dynamic typing guarantee? Is it really a sensible term? Motivate! (And explain what a type error is.)

What would the effects be with respect to static typing if a `method_missing` hook (similar to that in Ruby or the `doesNotUnderstand` method in some Smalltalk dialects) was added to the Java language (or equivalent, like C++ or C#). [A method missing hook is a method invoked on an object o when o is sent a message that it does not understand.]

4. Scripting? **(15%)**
Motivate why it is (or isn't) important for a *scripting language* to have literals for (lightweight) untyped (or generic/ parametrically polymorphic) simple data structures like lists, tuples and maps?

(You should probably start out by defining what a scripting language is.)

5. Weak/Strong vs. Dynamic/Static? **(15%)**
Explain and relate the concepts weak/strong typing and dynamic/static typing to each other. Give examples of languages that fit into the four different categories and explain and motivate your choices.