

6.001 Tutorial 12 Notes

TA: Gerald Dalley
09–10 May 2005

1 Announcements

- Last tutorial!
- Final exam: Johnson, May 19th, 9:00–12:00
- Problem set 11 (feedback) – please do it, we really do care!
- If you would like some office hours or a review session before the final, email me by this Wednesday so I can schedule it, reserve rooms, *etc.*

2 Asynchronous Computing

The whole idea of asynchronous computing is that two processes that have shared state might interact in different ways if they are running in parallel.

There are several reasons why we might want asynchronous computing. To speed up our programs, we might have each CPU work on a different part of our data. Secondly, we might actually want to be doing multiple different things at the same time. For example, in my computer vision research, I wrote an object-oriented class that has one “thread” continuously streaming live video from the internet and putting it into an image buffer. This data comes in whenever the network allows it. A separate thread does the number-crunching on the video to find moving objects and track them. These two pieces operate independently, except for the brief moments when the number-crunching portion snatches the most recent fully-downloaded video frame.

```
(define x 2)
(define (foo)
  (if (> x 0) (set! x (- x 1))))
(parallel-execute foo foo foo foo)
```

What outputs are possible if each `foo` is evaluated atomically?

What are the possible outputs if only primitives and special forms are atomic?

3 Review

Topics from this course:

- Scheme
- Procedures and recursion
- Orders of growth
- Data abstractions
- Higher-order procedures
- Program methodology
- Symbols and quotation
- Abstract data types
- Mutation
- Environment model
- Object-oriented systems
- Interpretation/evaluation
- Lazy evaluation
- Asynchronous computing

For the bulk of this tutorial, we’ll go through portions of the Fall 2002 Final Exam.

Final Feedback

1. How have you enjoyed/tolerated/hated 6.001?
2. What things have gone well?
3. What things have not?
4. What did you think of tutorial? Was it useful?
5. Are there any things you would have liked to have seen from me that would have helped?
6. If you could change anything about tutorial, what would it be?
7. Any other comments?

Thanks for your feedback!