

Homework 1: Introduction to Programming Concepts

Due: normal problems on Monday, August 27, 2007; extra credit problems on Wednesday, September 5, 2007.

In this homework you will learn some of the basics of Oz and the Mozart system [UseModels], and will get an overview of programming concepts [Concepts].

For all programming tasks, you must run your code using the Mozart/Oz system.

You must also provide evidence that your program is correct (for example, output from test cases). For each problem that requires code, turn in, using your WebCT account for this course, your code and your evidence that your code is correct (e.g., your testing code and its output).

For testing you may find the following procedure helpful.

```

proc {Assert B}
  if {Not B}
    then {Exception.raiseError assertionFailed}
  end
end

```

The `Assert` procedure can be used as in the following statement.

```
{Assert {Comb J I} == {CombB J I}}
```

Note that you would not use `Browse` around a call to `Assert`.

Don't hesitate to contact the staff if you are stuck at some point.

For background, you should have already read Chapter 1 of the textbook [RH04] (except section 1.7). But you may also want to refer to the reference and tutorial material on the Mozart/Oz web site <http://www.mozart-oz.org/>.

Conceptual Problems

- (5 points) [Concepts] [MapToLanguages]
What is the most important difference between a variable in Oz and a variable in C++ or Java?
- (5 points) [Concepts]
In Oz, what happens when a variable is used before it is bound? What is the technical term for this behavior?
- (5 points) [Concepts] [UseModels]
What is pattern matching used for in Oz? Given a brief example.
- (5 points) [Concepts]
According to chapter 1, why is programming with both cells and concurrency difficult? Have you seen enough evidence to agree or disagree with the book's position on this question?
- (20 points; extra credit) [Concepts]
Do problem 3 in chapter 1, program correctness.

Programming Problems

The following problems are from the textbook [RH04, section 1.18].

6. (20 points) [UseModels]
Do problem 2 in chapter 1, calculating combinations.
7. (10 points) [UseModels]
Do problem 5 in chapter 1, lazy evaluation.
8. (10 points; extra credit) [UseModels]
Do Problem 7 in chapter 1, explicit state.
9. (10 points; extra credit) [UseModels]
Do problem 8 in chapter 1, explicit state and functions.
10. (15 points; extra credit) [UseModels]
Do problem 10 in chapter 1, explicit state and concurrency.

References

- [RH04] Peter Van Roy and Seif Haridi. *Concepts, Techniques, and Models of Computer Programming*. The MIT Press, Cambridge, Mass., 2004.