CS480 - Final Exam
Due: Tuesday, 03/17/15, 11:59pm
(Absolutely NO Extensions and NO Late Work will be accepted!!!)

This is your final exam and your OWN work. This means closed neighbor, but not closed book/notes. Turn in your document using the TEACH website.

For this final, you can explore an area of compilers OR implement functions in IBTL. This final is not group work!!! Pick which of these you’d rather do: Write a Paper or Implement Functions

Write a Paper:
• You need to first begin by thinking of a topic and then doing a background literature review to learn more. I would suggest you use the ACM digital library for your search to see what has been published in this area. You will want to try to be as narrow as possible with your search, as the results can be a little overwhelming:) For instance, you might want to investigate compiler optimization, but this is going to return over 15,000 articles! Whereas, if you investigate compiler optimization using machine learning algorithms, it returns only about 2,000 articles. Use the Valley Library to access the ACM digital library: http://osulibrary.oregonstate.edu/research-databases

When reviewing a research paper, these are the areas one wants to consider:
  o Overview – what is the paper about, what was the author’s hypothesis, how did the author support or refute the research hypothesis, what are results/conclusions from the research, etc.

  o Critique – does the paper provide enough background for the reader, does the paper include ample proofs and examples to support the author’s claims, is the presentation of the paper easy to follow/understand, etc.

Read 2 research papers, and provide a 2-3 page 12-pt. single-spaced report for the review. Provide up to a half a page for a self-reflection and bibliography, i.e. no more than 4 pages total! You are required to provide an overview and critique for each research paper you chose to read. Please provide enough detail, and expand on the criteria provided for you. These topics are only a guideline, and you are free to expand on any of these in as much detail as you would like.

Implement Functions in IBTL:
• You can implement functions by adding this production:
  oper -> (:= name (name paramlist))
  letstmts -> (let ((funlist) (funtype)) exprlist)
  funlist -> name | name funlist
  funtype -> type | type funtype
  paramlist -> oper paramlist | ε
Here is an example definition of a function:

```c
int a(int b, int c) { … }
```

would be…

```lisp
(let ((a b c) (int int int)) …)
```

Here is an example call to the function:

```c
x = a(2, 3);
```

would be…

```lisp
(:= x (a 2 3))
```

Again, please turn in your document using the TEACH website by the due date!!!