CSC 301 – Fall 2015  
Final Exam  
Take Home Exam  
Due in Sakai Saturday 12/19/15 @ 11:55pm

Name __________________

100 points total

Part I, short answers, 5 points each, total 50 points

1. Name the stages of a typical programming language compiler?

2. Name at least one programming language for each of the following language paradigms:
   - imperative
   - applicative/functional
   - object-oriented
   - logic/rule-based
3. What do we mean by an “ambiguous grammar”?

4. What is meant by “higher-order programming” in functional languages?
5. What do we mean by a memory leak and why is it so dangerous?

6. What is the role of a data type in computing? And what is the formal definition of a data type?

7. What language feature does the use of a runtime stack in a programming language enable?
8. What does the following Prolog program do?

\[ p([X|Y], Y). \]

9. What is the difference between program translation and program interpretation?

10. What role does the **syntax** specification play in the definition of a programming language? What role does the **semantic** specification play in the definition of a programming language?
Part II -- Problems, 50 points total

1. This question refers to the following code fragment (10 points)

```c
int i = 1;
int j = 2;
f(i, j);
```

Function f is defined as:

```c
void f(int x, int y) {
    int max;
    if (x >= y)
        max = x;
    else
        max = y;
    x = max;
}
```

a) If arguments were passed **by value**, what would be the **final** value of the variables i and j after the call to f? Explain your answer.

b) If arguments were passed **by reference**, what would be the **final** value of the variables i and j after the call to f? Explain your answer.
2. Write an ML function `count` that given an integer list and an integer value will count the number of times the integer value appears in the list (hint: use recursion) (10 points).

Examples: `count([1,2,3,2],2) \rightarrow 2`
`count([1,2,3,2],3) \rightarrow 1`
`count([1,2,3,2],5) \rightarrow 0`
3. (10 points) Given the **Prolog** semantics in the appendix, informally show that

\[
\text{val1(plus(times(const(4),const(3)),const(1)), Q)}
\]

will evaluate to Q=13
4. Programming language syntax and semantics -- the following questions refer to the material given in the appendix to this exam. Justify your answers.

a) Show a parse tree and abstract syntax tree for the sentence (10 points):

   2 ♥ 3 ♣ 4

(The grammar and the abstract syntax is given in the appendix)
b) Using the Prolog semantic for Language Foo compute the semantic value for the expression.
(10 points):

\[ 2 \heartsuit 3 \clubsuit 4 \]
Appendix: Definitions for Question 3, Part II.

\[
\text{val1(plus}(X,Y),\text{Value}) : - \\
\text{\quad val1}(X,\text{XValue}), \\
\text{\quad val1}(Y,\text{YValue}), \\
\text{\quad Value is XValue + YValue.}
\]

\[
\text{val1(minus}(X,Y),\text{Value}) : - \\
\text{\quad val1}(X,\text{XValue}), \\
\text{\quad val1}(Y,\text{YValue}), \\
\text{\quad Value is XValue - YValue.}
\]

\[
\text{val1(times}(X,Y),\text{Value}) : - \\
\text{\quad val1}(X,\text{XValue}), \\
\text{\quad val1}(Y,\text{YValue}), \\
\text{\quad Value is XValue * YValue.}
\]

\[
\text{val1(const}(X),X).
\]
Appendix: Definitions for Question 4, Part II.

Grammar for Language Foo

```plaintext
<addexp>* ::= <addexp> ♠ <mulexp>
   | <mulexp>

<mulexp> ::= <mulexp> ♥ <rootexp>
   | <rootexp>

<rootexp> ::= (<addexp>)
   | <constant>

<constant> ::= any valid integer constant
```

Abstract Syntax for Language Foo

```plaintext
<expr> ::= club(<expr>,<expr>)
   | heart(<expr>,<expr>)
   | const(<const>)

<const> ::= any valid integer constant
```

Prolog Semantics for Language Foo

```prolog
val(club(X,Y),Value) :-
   val(X,XValue),
   val(Y,YValue),
   Value is XValue + YValue.

val(heart(X,Y),Value) :-
   val(X,XValue),
   val(Y,YValue),
   Value is XValue * YValue.

val(const(X),X).
```