Exercises—Formal Grammars

Part 1, due Thursday Sept. 15

1) Discuss these as a team, and bring proposed answers to class today. For the language of your compiler project:

Which of these are syntactically legal in some appropriate context? Why or why not?

int test;
float test1, test2;
int i = 0;
int abc[10];
int *p;
int fibonacci ( int n );
a = b = 0;
return ( result );
cin >> x >> y;

Part 2, due Tuesday Sept. 20

1) Modify the grammar you were given in class for our language subset to permit “void” as a function return type, and to allow “return” to be used without a return value. Include any new or modified grammar rules that this would require, as well as definitions of any new tokens.

2) Modify the grammar from exercise #1 above to permit the single-character token “&” in formal parameter lists between a type name and parameter name. In real C++, this would represent passing the parameter by reference; in our subset, it has no semantic effect since that is already the default passing mechanism. Include any new or modified grammar rules that this would require, as well as definitions of any new tokens.

Part 3, due at the end of the term

As a team, modify the grammar of Part 3 to permit “if” statements that do not have an “else” clause. See the discussion of the “dangling else ambiguity” in your two textbooks.

Your final project must incorporate the changes from Parts 2 and 3