Michael Smith

CSE 320 Programming Languages

October 17, 2018

Exercises 1-3

1. Some would consider C++ is a multi-paradigm language which includes imperative and object-oriented because of how close C++ is to the C language. However, they would be classified differently because C++ does not rely on the procedural programming to function. You can create classes which are standalone. Yes, I believe that an object-oriented language can be programmed in imperative style, but it will lose some of the deep functionality that the language will offer. However, the reverse is not correct. C would not be able to create classes, or objects. Or be able to handle polymorphism or inheritance.
2. Let:

D -> Declaration

VL -> Variable List

Ty -> type

1. Declaration in the grammar is type followed by list of variables
2. Type in the grammar is one of the three: int, float, or char
3. A variable list can be single variable, single array, single variable with comma followed by a list of variables or single array declaration with a comma followed by a list of variables

So then we have the grammar:

D -> TyVL

Ty -> int | float | char

VL -> id | id[num] | id VL | id[num], VL

1. Ambiguous means that there are two different parse trees possible for a single statement. Or that we can derive the same statement in two ways. While if the grammar is unambiguous then there is only one unique way to generate any string. An example would be: if E then if E then id=id else S. Here the ambiguity is that we can not be sure to which if the else is associated with.