Meaning of Identifiers

**Disambiguating Names**

A Cb identifier is disambiguated mainly by the following characteristics: lexical uniqueness and function scope

**Lexical Uniqueness**

Cb identifiers are created with a combination of Latin characters and the underscore character as specified (and constrained) in the section, *Lexical Conventions*. All identifiers are first disambiguated by its lexical name being different from all other identifiers in the file.

**Function Scope**

Cb identifiers have nearly no scope (brackets have no effect on the life of a variable); all variables are global to the file. However, a single exception is made to any variables declared within a method to avoid the unintentional manipulation of values. For example:

Method Note test()  
{  
 Note n = (D, duration2);  
 return n;  
}

Note n = (C, duration);

doremi();

Here, a note is declared twice; once while defining a function and once after defining the method. Moreover, the method is subsequently called after the second declaration. In this scenario, having function scope is important since it is likely that the user does not want to let the method call alter the declaration/initialization of identifiers he/she makes beforehand.

**Object Types**

Cb supports the following fundamental data types: Int, Note, Chord, Scale, Stanza, and Score.

These types are defined in further detail in

any identifiers declared within the scope of a function

However, identifiers being accessed from an external file are accessed via the namespace of t