## Sets versus Lists

Sets are a formally defined mathematical concept. As such we have well defined rules for sets such as not putting duplicate elements into a set. Sets also have operations such as union, intersection, and complement.

A list, on the other hand, is a collection of things but a collection without the formal rules of sets. For example, a list might have repeated items. Or, in some lists, the order of the items may be important. There are many times in the programming environment when we want to work with a set and there are many times when we want to work with a list. Often, even in the same problem, we have instances of both sets and lists. For example, in representing a class of students we may have the set of names {Alan B., Tom G., Betty R., Alan T., Sue V., Ann G., William T., Betty L., Sue B., Janet K}. This also means that we have the set of first names {Alan, Tom, Betty, Sue, Ann, William, Janet} but then too we have the list of first names [Alan, Tom, Betty, Alan, Sue, Ann, William, Betty, Sue, Janet]. Each student has an age so the list of ages might be [19, 20, 32, 19, 20, 20, 35, 20, 19, 21] which means that the set of ages is {19, 20, 21, 32, 35}. Note that the list of ages might well be tied to the list of first names and if so then it would be important to keep the order of items in the lists parallel.

In programming problems it is essential to recognize when you are dealing with sets and when you are dealing with lists. Naturally, common practice often undermines the true definition of a set. It is quite common for people to talk about a "data set" when they are really talking about a list of values. Just because someone calls something a "data set" does not mean that it is really a set. It may be a list.