To begin with, my design starts with the Instruments.java file. This file contains a single public class called instruments. There is no constructor for this class because you shouldn't be able to have an "Instrument" object. In this class, I decided that the only attributes that all instruments share are their names and what family they belong to in their classification of instruments. The only method that all instruments share is to be played, so I defined a method for each instrument to return their name when they are "played".

From this point, there are only 4 more singular files. Each file contains a public class for each type, Brass, Woodwind etc. In each file there are then classes for each instrument of that respective type. In terms of extensibility, this means that to add a new instrument, a programmer would simply have to define a new class following the format that I have used for each instrument.

In the Brass.java file, I have given the Brass class 2 attributes, which means all Brass instruments will share these attributes. From a cursory knowledge of musical instruments, I know that all brass instruments are created out of brass tubing, and so each instrument will have a tube length attribute. They also can be separated into two groups with an attribute called "Bore". With these attributes defined, every brass instrument will inherit them.

Each instrument is defined following a formula. A class is defined with the name of the instrument, and the class will extend the superclass that it belongs to. For example:

class Trumpet extends Brass

The class then has its most specific attributes defined. In the case of Brass instruments, not all instruments have valves, only those of the "Valved" family. It is at this point that some unavoidable redundancy occurs, because you need to redefine the "valves" attribute with every class. A solution for this could be to add an extra hierarchical class between the instrument type and the instruments themselves, which would be the family that the instrument belongs to. In the constructor for every single instrument, the developer needs to set the attributes "name" and "family". The name will be the name of the instrument, so in the case of the trumpet the name would be "Trumpet" and the family attribute is the classification of that type of instrument that the instrument belongs to. In the case of trumpet, it is a "Valved" brass instrument. Each instrument can then have its most specific attributes defined.

Having the constructors only in the bottom level classes provides an advantage because in real life you couldn't have a "Brass" object, it would have to be a Trumpet or a Tuba or any other brass instrument.

Notably, percussion instruments are highly variable by nature, and so the percussion class contains no attributes that I could find that are specific to only percussion instruments (and not all instruments).