

1) The Song class contains two equals() methods. Explain the difference using your knowledge about inheritance.

The first method will work for arguments belonging to the class Song, the second method will work for arguments of any subclass of object (So objects of any class), however to actually properly make the comparison these objects must still possess the title and instrument fields of song. As such for the function to work as intended the argument can be any superClass of Song defined in the following manner Object o = new Song.

2) Why do we need the version of equals() that takes an Object as the parameter in this class?

Because of the way I structured my code, equals is only called on an object belonging to class song once. When this happens the argument is guaranteed to be of class song, so this second version of equals() is never called. This is largely because when an error occurs in reading input, I set the Song object to null, then check for null songs before I reach the equals() operation. If you were instead to keep invalid values in your collection you could certainly make use of the second equals to handle those errors. There are many ways to create code to solve the same problem and as such I am sure there are many solutions to this problem that use this second equals() to their advantage in a variety of creative ways in addition to the way I suggested.