

```

package GenderMag2;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Random;
public class Student1 {
    public static void main(String[] args) {
        // Open the input and output files
        try (BufferedReader reader = new BufferedReader(new
FileReader("songs.txt"));
            BufferedWriter writer = new BufferedWriter(new
FileWriter("songsUpdated.txt"))) {
            // Header Row
            writer.write("SongID,Song Name,Artist,Album,Year\n");

            Random random = new Random();
            String line;
            while ((line = reader.readLine()) != null) {
                // Splits each line of the text file into columns (So that it's like a table)
                String[] columns = line.split(",");
                // Generating song ID's for every song
                int songID = 1000000 + random.nextInt(9000000);
                // Writing the song ID's to be in front of the other columns
                writer.write(songID + "," + columns[0] + "," + columns[1] + "," +
columns[2] + "," + columns[3] + "\n");
            }
            System.out.println("Successfully updated");
        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }
}

```

//

```

package GenderMag2;
public class Song<songData1, songData2, songData3, songData4> {
    private int songID;
    private songData1 songData1;

```

```
private songData2 songData2;  
private songData3 songData3;  
private songData4 songData4;  
public Song(int songID, songData1 songData1, songData2 songData2, songData3  
songData3, songData4 songData4) {  
    this.songID = songID;  
    this.songData1 = songData1;  
    this.songData2 = songData2;  
    this.songData3 = songData3;  
    this.songData4 = songData4;  
}  
public int getSongID() {  
    return songID;  
}  
public void setSongID(int songID) {  
    this.songID = songID;  
}  
public songData1 getSongData1() {  
    return songData1;  
}  
public void setSongData1(songData1 songData1) {  
    this.songData1 = songData1;  
}  
public songData2 getSongData2() {  
    return songData2;  
}  
public void setSongData2(songData2 songData2) {  
    this.songData2 = songData2;  
}  
public songData3 getSongData3() {  
    return songData3;  
}  
public void setSongData3(songData3 songData3) {  
    this.songData3 = songData3;  
}  
public songData4 getSongData4() {  
    return songData4;  
}  
public void setSongData4(songData4 songData4) {  
    this.songData4 = songData4;  
}
```

```

    }
    @Override
    public String toString() {
        return "Song:" + songID + ", " + songData1 + ", " + songData2 + ", " + songData3 +
        ", " + songData4 + "";
    }
}

//

public static <Data1, Data2, Data3, Data4> void displayAllSongs(LinkedList<Song<Data1,
Data2, Data3, Data4>> allSongs) {
    for (Song<Data1, Data2, Data3, Data4> songs : allSongs) {
        System.out.println("Song ID: " + songs.getSongID());
        System.out.println("Song Data 1: " + songs.getSongData1());
        System.out.println("Song Data 2: " + songs.getSongData2());
        System.out.println("Song Data 3: " + songs.getSongData3());
    }
}
/*This is all purely hypothetical because I couldn't get the main method to work on my
generic class

```

```

        Song<String, Integer, Double, Double> firstSong = new Song<>("Hello", 11, 11.3,
16.4);
        Song<Integer, Integer, Boolean, Boolean> secondSong = new Song<>(239,12, false,
true);
        Song<Boolean, Boolean, Integer, String> thirdSong = new Song<>(false, false, 4 ,
"Hello");

```

```

//Tests getters
System.out.println(firstSong.getSongID());
System.out.println(firstSong.getSongData1());
System.out.println(firstSong.getSongData2());
System.out.println(firstSong.getSongData3());

System.out.println(secondSong.getSongID());
System.out.println(secondSong.getSongData1());
System.out.println(secondSong.getSongData2());
System.out.println(secondSong.getSongData3());

System.out.println(thirdSong.getSongID());

```

```
System.out.println(thirdSong.getSongData1());
System.out.println(thirdSong.getSongData2());
System.out.println(thirdSong.getSongData3());
//Tests setters
```

```
        firstSong.setSongData1("Bruh");
firstSong.setSongData2(34);
firstSong.setSongData3(3.4);
```

```
secondSong.setSongData1(4);
secondSong.setSongData2(5);
secondSong.setSongData3(false);
```

```
thirdSong.setSongData1(true);
thirdSong.setSongData2(true);
thirdSong.setSongData3(1);
```

```
//Tests toString
```

```
System.out.println(firstSong.toString());
System.out.println(secondSong.toString());
System.out.println(thirdSong.toString());
```

```
*/
```

(Referencing from a previous task)

```
while ((line = reader.readLine()) != null) {
    // Splits each line of the text file into columns (So that it's like a table)
    String[] columns = line.split(",");

    Song<String,String,Integer,Double,Double> newSong = new
Song<>(songID,
columns[0],Integer.parseInt(columns[1]),Double.parseDouble(columns[2]),Double.parseDo
uble(columns[3]));

    songs.add(newSong);

}
}
```

```
Song<songID,String,Integer,Double,Double>[] myArray = new Song[songs.size()];
```

```
songs.toArray(myArray);  
return myArray;
```

In the case of [LinkedList](#) we would replace the line:

```
public static Song<String, String, Integer, Double>[] readSongsToArray(String filename) {  
    List<Song<String, String, Integer, Double>> songList = new ArrayList<>();  
}
```

With

```
public static Song<String, String, Integer, Double>[] readSongsToArray(String filename) {  
    LinkedList<Song<String, String, Integer, Double>> songList = new  
    LinkedList<>();  
}
```