**Report Project 2: Implementation of FileList**

FileList implementation require to use RandomAccessFile (Binary) for storing relations in a disk to maintain persistence. Class FileList consists of two main methods: add and get that were left to be implemented. Implementation of add require tuple to be packed into an fixed length byte[] that is equal to recordsize and then its written on the binary file. Get reads the byte[] and its finally unpacked to return the tuple.

* **Get\_recordsize(Comparable[] tuple):** Added to Table.java. It returns the length of the recordsize required for particular instance of Table class. It checks the “type” of all the attribute and according to that recordsize is calculated.
* **Pack(Comparable[] tuple):** Added to FileList.java. It uses methods that convert a particular “Class” type to a Byte[] and finally pack the tuple into record.
* **Unpack(byte[] byte):** Added to FileList.java. Unpack the byte[] and converts it to Comparable[] tuple.
* **Add(Comparable[] tuple) :** Added to FileList.java. Add passed tuple to the file.
* **Get(int index):** Added to FileList.java. Reads the byte[] from a file and return the tuple of relation.

**Note:**

FileList is completely implemented with all the required methods. Relations can be stored into file and therefore can be extracted, however what remains is to use it in place of ArrayList.

Creating FileList object can be made separately in MovieDB class to mirror the required functionalities.