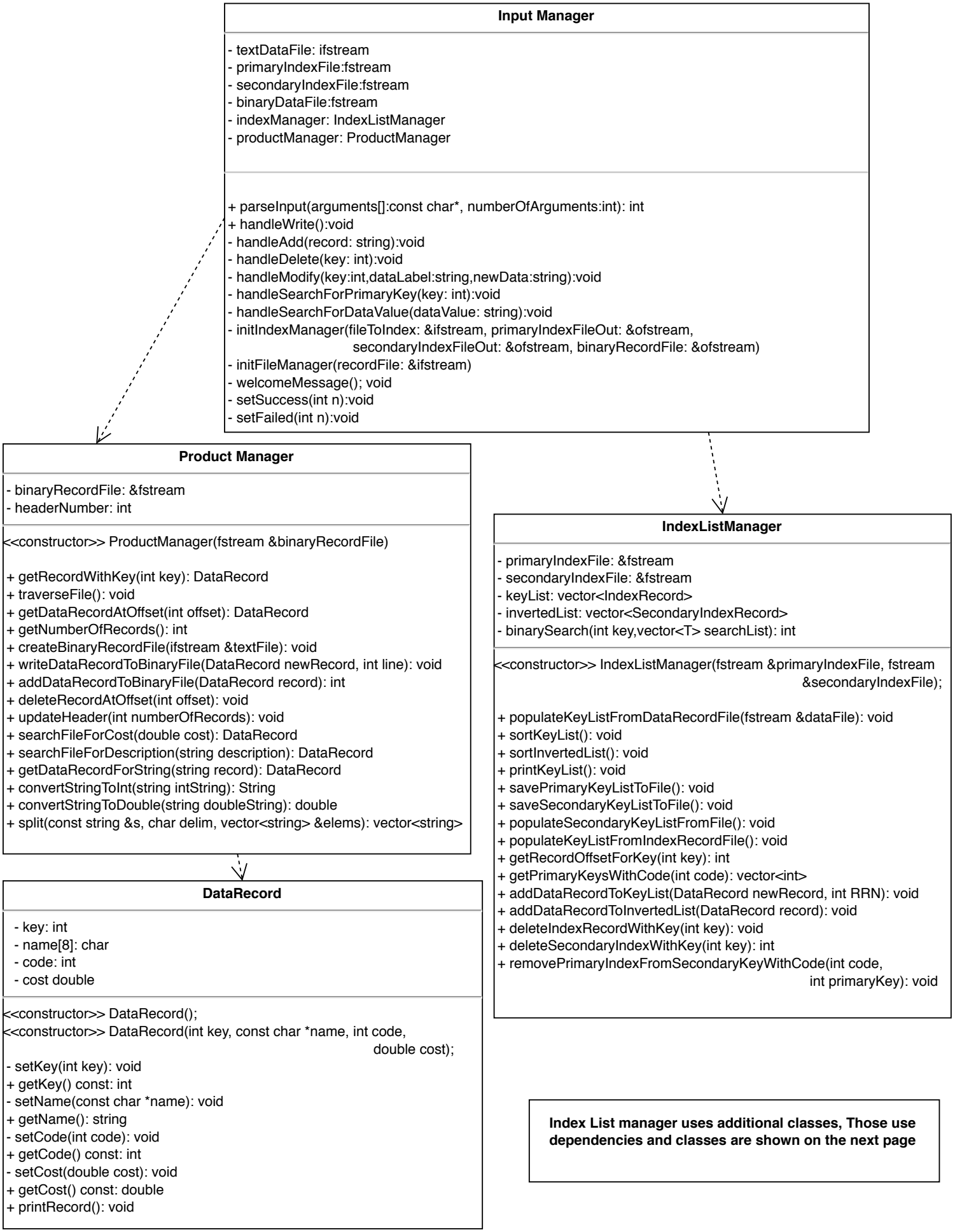
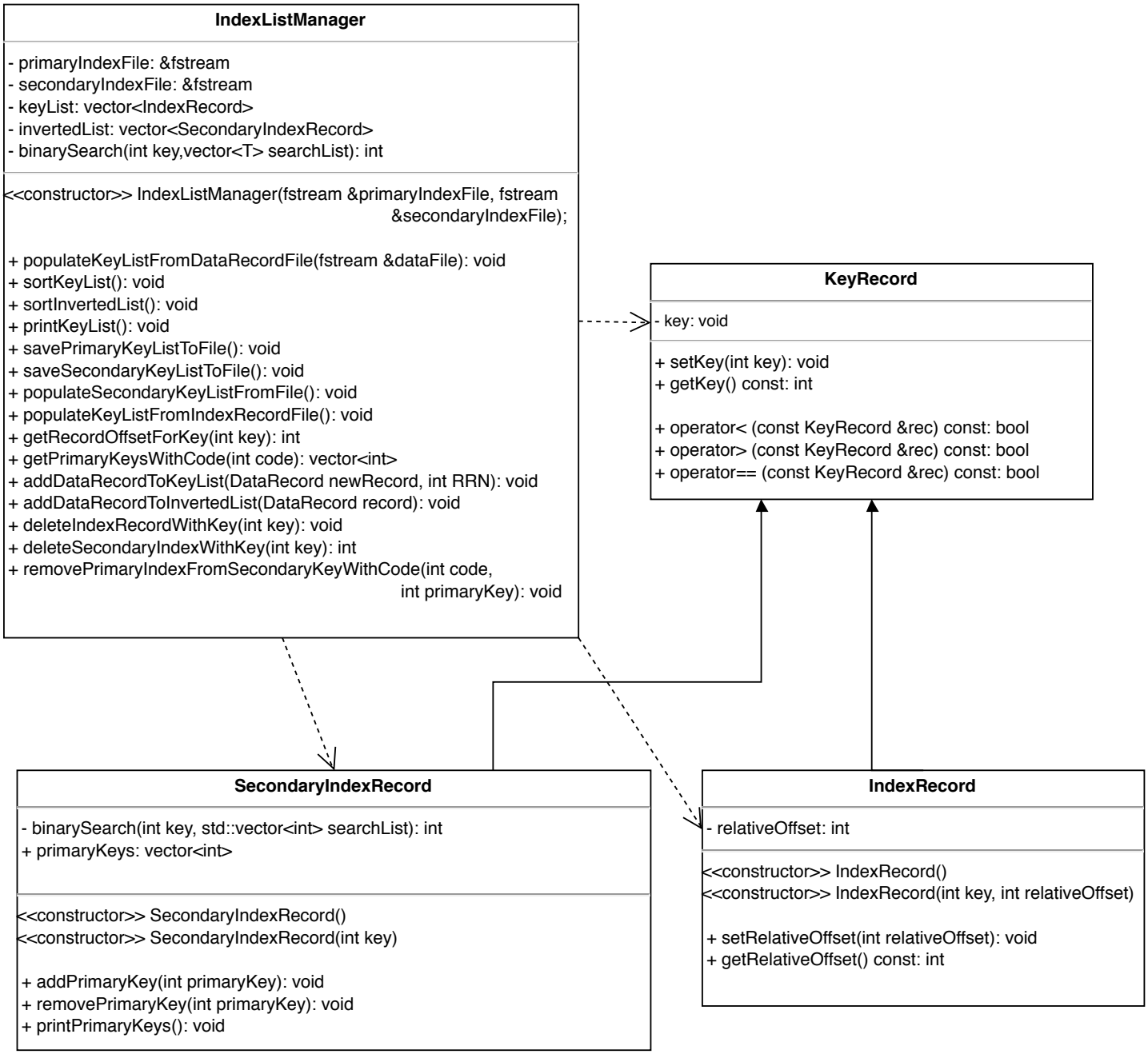

Input Manager
- textDataFile: ifstream
- primaryIndexFile:fstream
- secondaryIndexFile:fstream
- binaryDataFile:fstream
- indexManager: IndexListManager
- productManager: ProductManager
+ parseInput(arguments[]:const char*, numberOfArguments:int): int
+ handleWrite():void
- handleAdd(record: string):void
- handleDelete(key: int):void
- handleModify(key:int,dataLabel:string,newData:string):void
- handleSearchForPrimaryKey(key: int):void
- handleSearchForDataValue(dataValue: string):void
- initIndexManager(fileToIndex: &ifstream, primaryIndexFileOut: &ofstream,
secondaryIndexFileOut: &ofstream, binaryRecordFile: &ofstream)
- initFileManager(recordFile: &ifstream)
- welcomeMessage(); void
- setSuccess(int n):void
- setFailed(int n):void
Product Manager
- binaryRecordFile: &fstream
- headerNumber: int
<<constructor>> ProductManager(fstream &binaryRecordFile)
+ getRecordWithKey(int key): DataRecord
+ traverseFile(): void
+ getDataRecordAtOffset(int offset): DataRecord
+ getNumberOfRecords(): int
+ createBinaryRecordFile(ifstream &textFile): void
+ writeDataRecordToBinaryFile(DataRecord newRecord, int line): void
+ addDataRecordToBinaryFile(DataRecord record): int
+ deleteRecordAtOffset(int offset): void
+ updateHeader(int numberOfRecords): void
+ searchFileForCost(double cost): DataRecord
+ searchFileForDescription(string description): DataRecord
+ getDataRecordForString(string record): DataRecord
+ convertStringToInt(string intString): String
+ convertStringToDouble(string doubleString): double
+ split(const string &s, char delim, vector<string> &elems): vector<string>
IndexListManager
- primaryIndexFile: &fstream
- secondaryIndexFile: &fstream
- keyList: vector<IndexRecord>
- invertedList: vector<SecondaryIndexRecord>
- binarySearch(int key,vector<T> searchList): int
<<constructor>> IndexListManager(fstream &primaryIndexFile, fstream
&secondaryIndexFile);
+ populateKeyListFromDataRecordFile(fstream &dataFile): void
+ sortKeyList(): void
+ sortInvertedList(): void
+ printKeyList(): void
+ savePrimaryKeyListToFile(): void
+ saveSecondaryKeyListToFile(): void
+ populateSecondaryKeyListFromFile(): void
+ populateKeyListFromIndexRecordFile(): void
+ getRecordOffsetForKey(int key): int
+ getPrimaryKeysWithCode(int code): vector<int>
+ addDataRecordToKeyList(DataRecord newRecord, int RRN): void
+ addDataRecordToInvertedList(DataRecord record): void
+ deleteIndexRecordWithKey(int key): void
+ deleteSecondaryIndexWithKey(int key): int
+ removePrimaryIndexFromSecondaryKeyWithCode(int code,
int primaryKey): void
DataRecord
- key: int
- name[8]: char
- code: int
- cost double
<<constructor>> DataRecord();
<<constructor>> DataRecord(int key, const char *name, int code,
double cost);
- setKey(int key): void
+ getKey() const: int
- setName(const char *name): void
+ getName(): string
- setCode(int code): void
+ getCode() const: int
- setCost(double cost): void
+ getCost() const: double
+ printRecord(): void
Index List manager uses additional classes, Those use dependencies and classes are shown on the next page


IndexListManager
- primaryIndexFile: &fstream
- secondaryIndexFile: &fstream
- keyList: vector<IndexRecord>
- invertedList: vector<SecondaryIndexRecord>
- binarySearch(int key,vector<T> searchList): int
<<constructor>> IndexListManager(fstream &primaryIndexFile, fstream &secondaryIndexFile);
+ populateKeyListFromDataRecordFile(fstream &dataFile): void
+ sortKeyList(): void
+ sortInvertedList(): void
+ printKeyList(): void
+ savePrimaryKeyListToFile(): void
+ saveSecondaryKeyListToFile(): void
+ populateSecondaryKeyListFromFile(): void
+ populateKeyListFromIndexRecordFile(): void
+ getRecordOffsetForKey(int key): int
+ getPrimaryKeysWithCode(int code): vector<int>
+ addDataRecordToKeyList(DataRecord newRecord, int RRN): void
+ addDataRecordToInvertedList(DataRecord record): void
+ deleteIndexRecordWithKey(int key): void
+ deleteSecondaryIndexWithKey(int key): int
+ removePrimaryIndexFromSecondaryKeyWithCode(int code, int primaryKey): void
KeyRecord
- key: void
+ setKey(int key): void
+ getKey() const: int
+ operator< (const KeyRecord &rec) const: bool
+ operator> (const KeyRecord &rec) const: bool
+ operator== (const KeyRecord &rec) const: bool
SecondaryIndexRecord
- binarySearch(int key, std::vector<int> searchList): int
+ primaryKeys: vector<int>
<<constructor>> SecondaryIndexRecord()
<<constructor>> SecondaryIndexRecord(int key)
+ addPrimaryKey(int primaryKey): void
+ removePrimaryKey(int primaryKey): void
+ printPrimaryKeys(): void
IndexRecord
- relativeOffset: int
<<constructor>> IndexRecord()
<<constructor>> IndexRecord(int key, int relativeOffset)
+ setRelativeOffset(int relativeOffset): void
+ getRelativeOffset() const: int