

# Project 1: KVStore using Log-structured Merge Tree

The handout files include two main parts:

- The `KVStoreAPI` class in `kvstore_api.h` that specifies the interface of KVStore.
- Test files including correctness test (`correctness.cc`) and persistence test (`persistence.cc`).

Explanation of each handout file:

```
.
├─ Makefile // Makefile if you use GNU Make
├─ README.md // This readme file
├─ correctness.cc // Correctness test, you should not modify this file
├─ data // Data directory used in our test
├─ kvstore.cc // your implementation
├─ kvstore.h // your implementation
├─ kvstore_api.h // KVStoreAPI, you should not modify this file
├─ persistence.cc // Persistence test, you should not modify this file
├─ utils.h // Provides some cross-platform file/directory interface
├─ MurmurHash3.h // Provides murmur3 hash function
└─ test.h // Base class for testing, you should not modify this file
```

First have a look at the `kvstore_api.h` file to check functions you need to implement. Then modify the `kvstore.cc` and `kvstore.h` files and feel free to add new class files.

We will use all files with `.cc`, `.cpp`, `.cxx` suffixes to build correctness and persistence tests. Thus, you can use any IDE to finish this project as long as you ensure that all C++ source files are submitted.

For the test files, of course you could modify it to debug your programs. But remember to change it back when you are testing.

Good luck :)