

# 一、answer all questions specified above

## Part 1a: Simple Client

Think about how to collect statistics from multiple threads.

在管理多个 thread 的类（即主类）中建立 HashMap 用于存储，在进程开始时将 HashMap 作为形式参数传入在线程内修改来收集多线程的数据。

## Part 1b: Sequential and Per-thread HTTP Servers

how to handle multiple threads reading and adding to the Map.

在管理多个 thread 的进程中建立 HashMap 用于存储，层层传递，在创建 request 对象时初始化类中的 HashMap 进行管理。

```
WebRequestHandler wrh =
    new WebRequestHandler( connectionSocket, cfgMap, fileCache );

public WebRequestHandler(Socket connectionSocket,
    Map<String, String> cfgMap, Map<String, String> fileCache)
throws Exception
{
    reqCount ++;
    this.fileCache = fileCache;
    ...
}
```

Heartbeat monitoring: Please describe a particular design and implement it.

## Part 2b (Option 1): Async Server: Multiplexed, Nonblocking Server (Reactive Server)

- **Design question:** a production-level server should have a timeout thread. Upon accepting a new connection, the accept handler should register a timeout event with the timeout thread with a callback function. The timeout value is specified by `IncompleteTimeout`. The default timeout value is 3 seconds. If the connection does not give a complete request to the server approximately within timeout from the time of being accepted, the server should disconnect the connection. Note that the timeout monitoring thread should not directly close a channel that the dispatcher thread is still monitoring (why?). You need to think very carefully about the exact details of the interaction between these two threads, and describe your software design in your final report. You DO not need to implement it, but if you do, please let us know in your report, and you will receive a 5% bonus points if your implementation pass the functionality test.