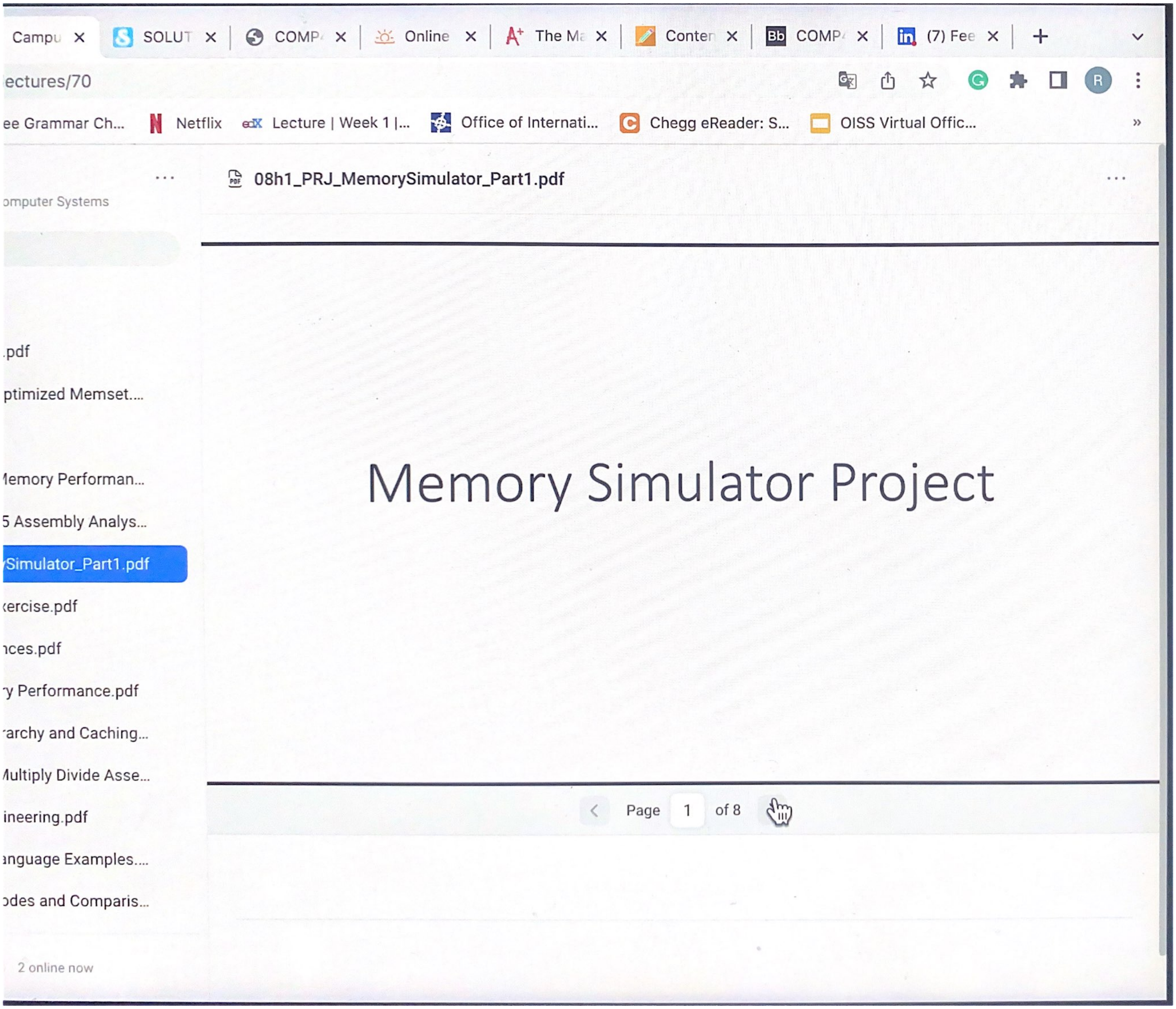
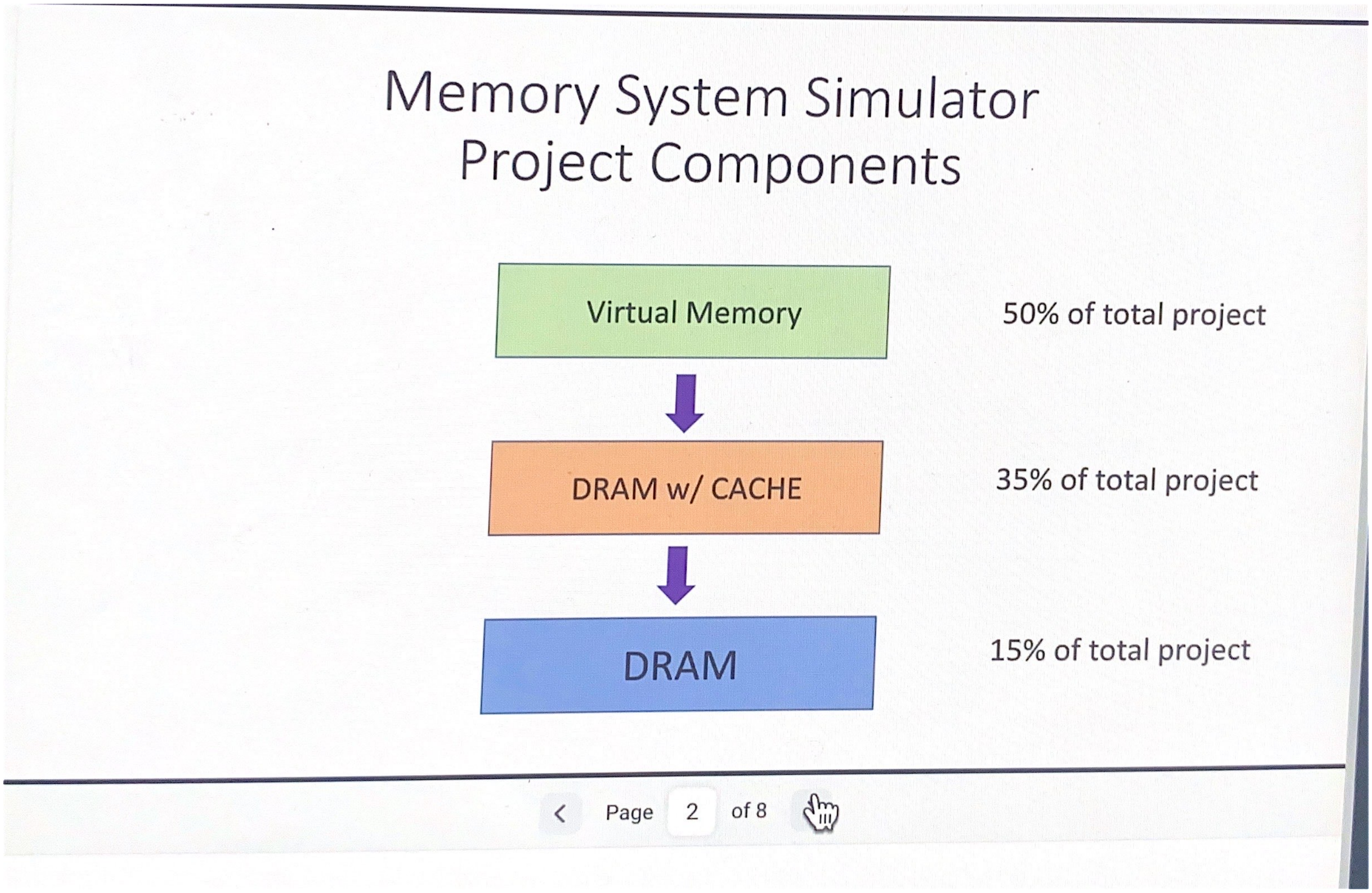


Memory Simulator Project

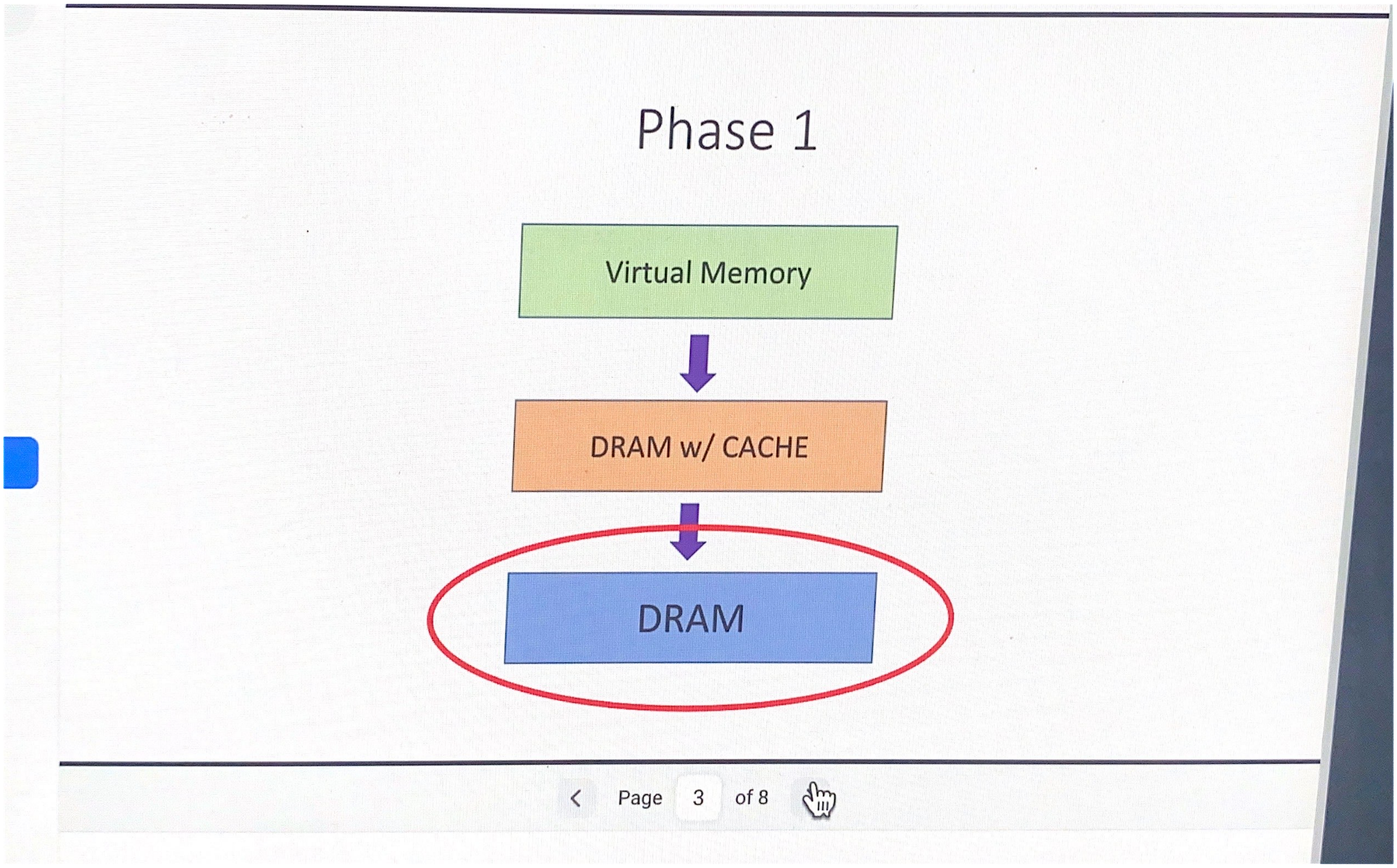


Memory System Simulator
Project Components

Virtual Memory — 50% of total project

DRAM w/ CACHE — 35% of total project

DRAM — 15% of total project

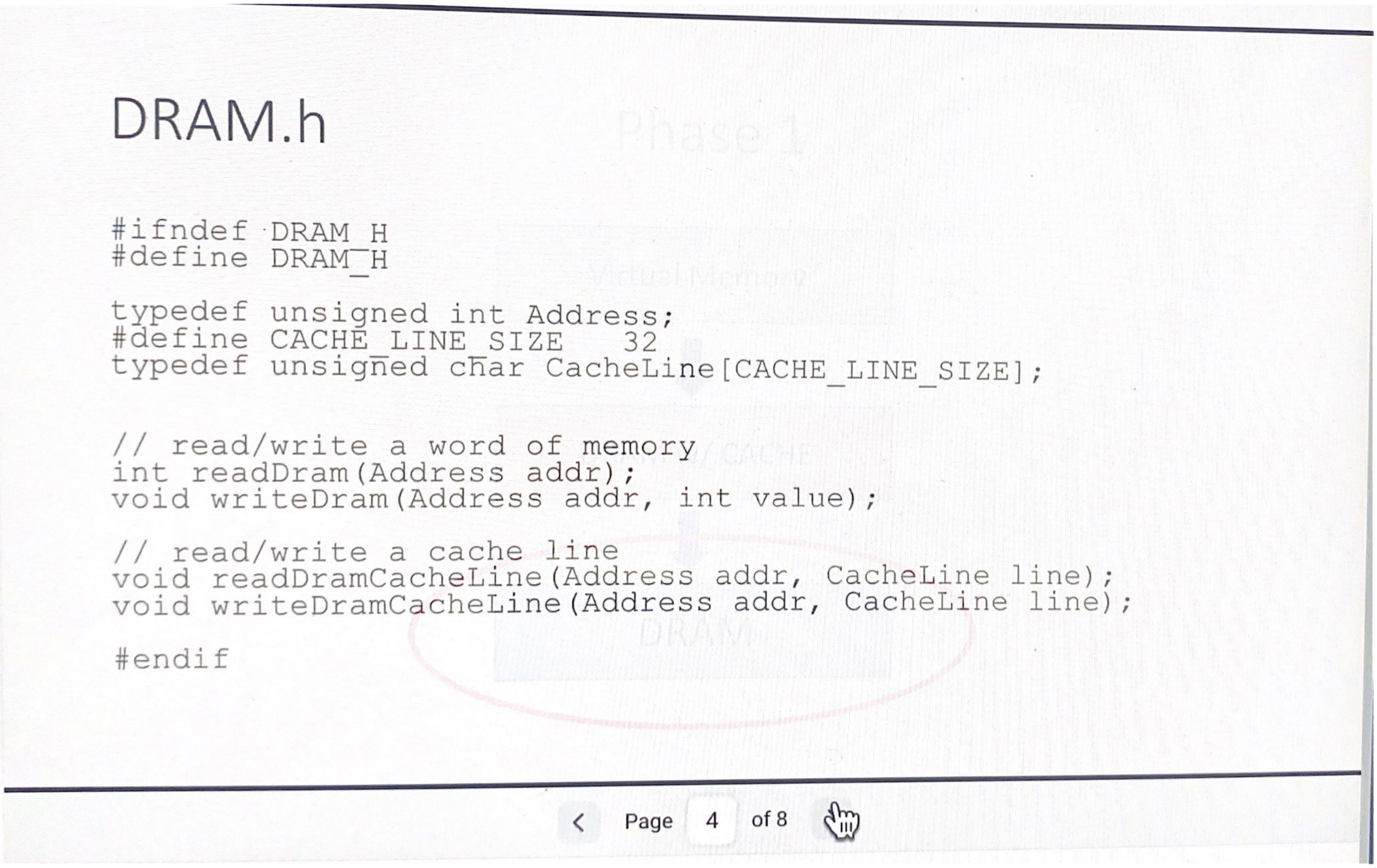


Phase 1

Virtual Memory

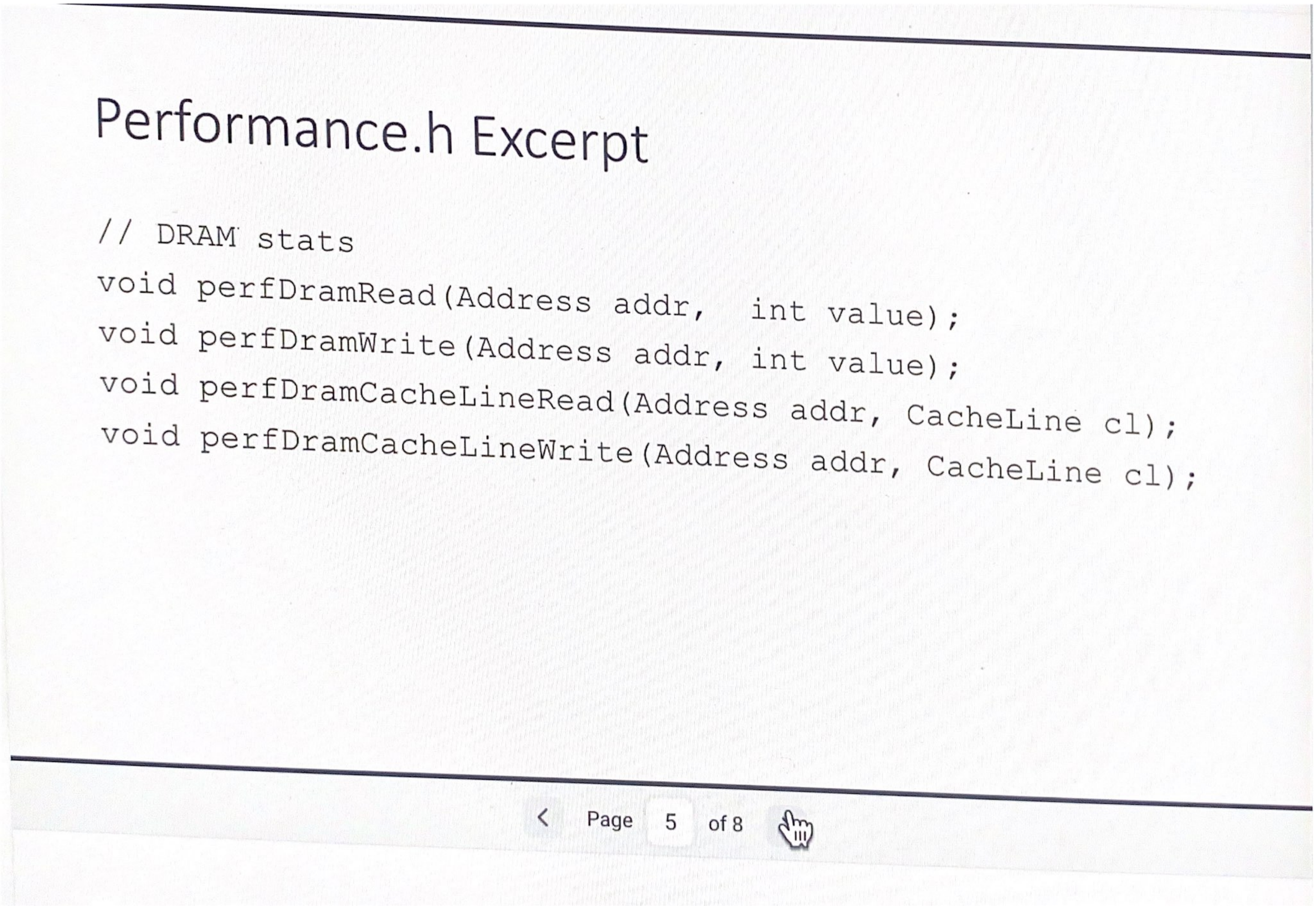
DRAM w/ CACHE

DRAM



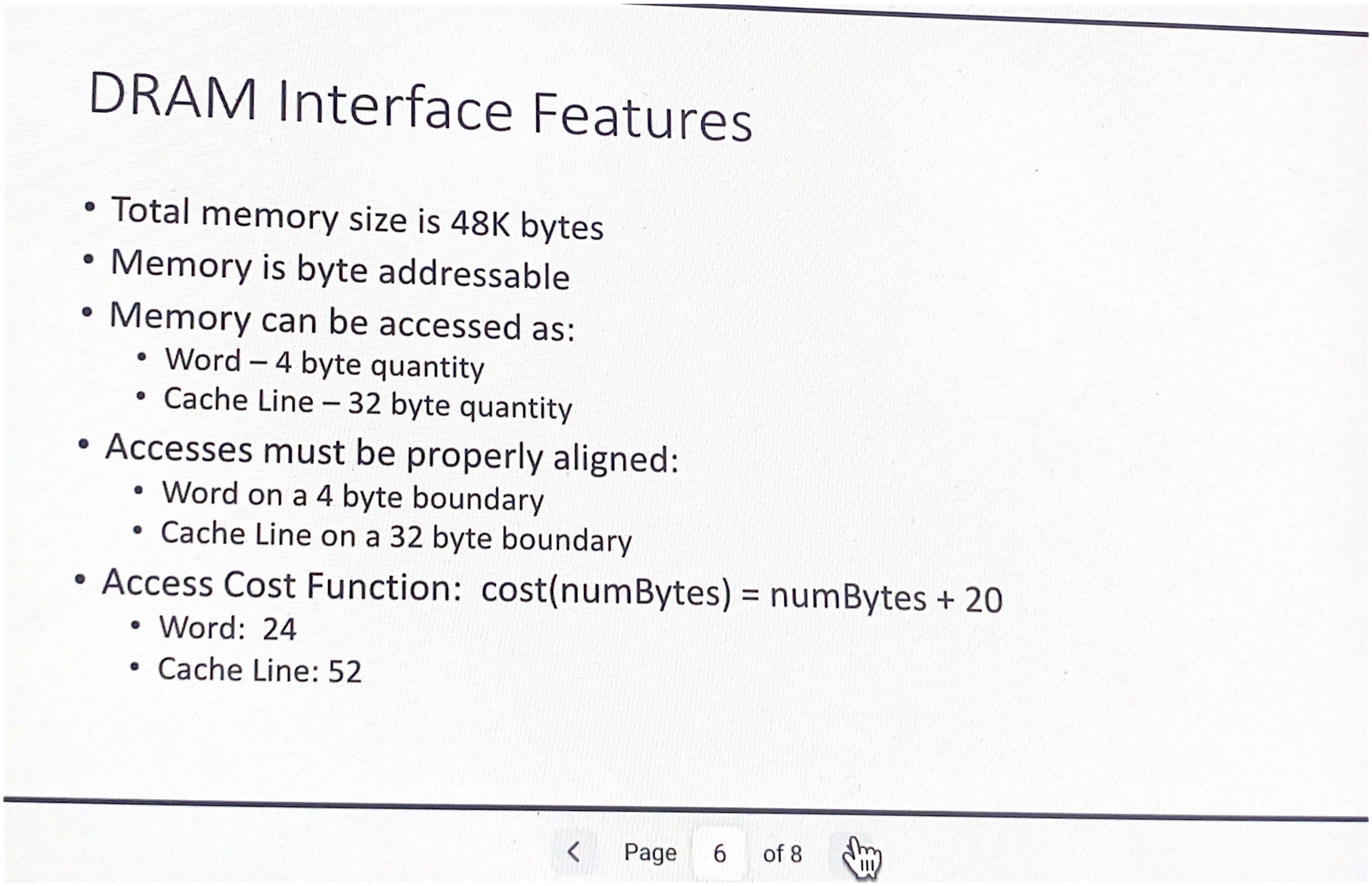
DRAM.h

```
#ifndef DRAM_H
#define DRAM_H

typedef unsigned int Address;
#define CACHE_LINE_SIZE   32
typedef unsigned char CacheLine[CACHE_LINE_SIZE];


// read/write a word of memory
int readDram(Address addr);
void writeDram(Address addr, int value);

// read/write a cache line
void readDramCacheLine(Address addr, CacheLine line);
void writeDramCacheLine(Address addr, CacheLine line);

#endif
```



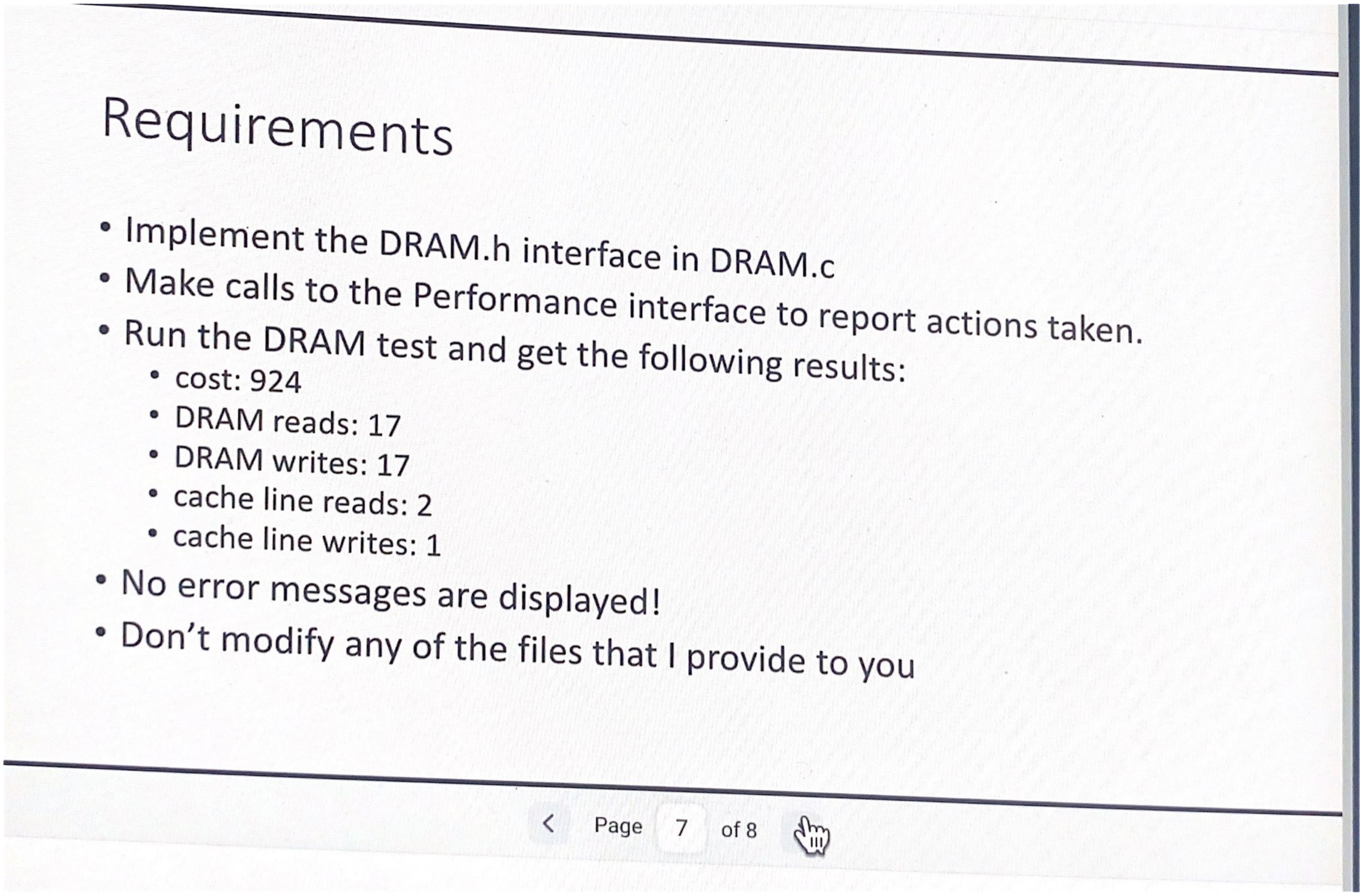
Performance.h Excerpt

```
// DRAM stats
void perfDramRead(Address addr,  int value);
void perfDramWrite(Address addr, int value);
void perfDramCacheLineRead(Address addr, CacheLine cl);
void perfDramCacheLineWrite(Address addr, CacheLine cl);
```



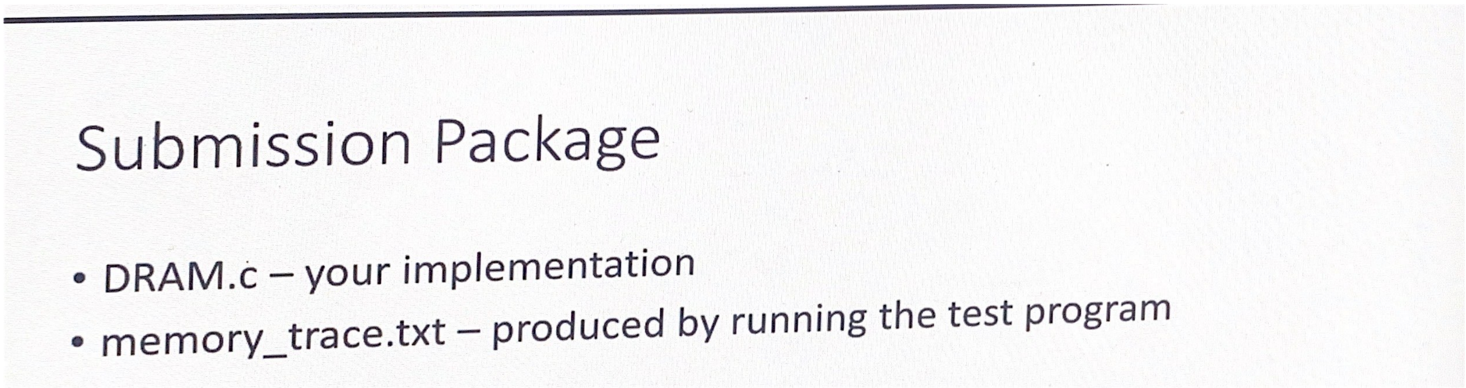
DRAM Interface Features

- Total memory size is 48K bytes
- Memory is byte addressable
- Memory can be accessed as:
  - Word – 4 byte quantity
  - Cache Line – 32 byte quantity
- Accesses must be properly aligned:
  - Word on a 4 byte boundary
  - Cache Line on a 32 byte boundary
- Access Cost Function: cost(numBytes) = numBytes + 20
  - Word: 24
  - Cache Line: 52



Requirements

- Implement the DRAM.h interface in DRAM.c
- Make calls to the Performance interface to report actions taken.
- Run the DRAM test and get the following results:
  - cost: 924
  - DRAM reads: 17
  - DRAM writes: 17
  - cache line reads: 2
  - cache line writes: 1
- No error messages are displayed!
- Don’t modify any of the files that I provide to you



Submission Package

- DRAM.c – your implementation
- memory_trace.txt – produced by running the test program