CS 448
Project 2

Team:
Justin Jaworski
Manmohit Sehgal

Note :
-- Everything compiles and runs with the given make command
-- In Part 1. We had an error regarding Xlint but that doesnot alter or change the compliation and the output in anyway.

Part 1 Implementation:
– Rather than having one big file, We broke up the code into separate files to make it easy to read and implement.
– We Implemented the following files:

a) HashTable.java
- – Contains method for hashtable which are used in the BufMgr.java

b) myDescriptionClass.java
- – Contains helper methods which are used in hashtable.

c) descriptions
- – Contains helper methods which are used in BufMgr

Part 1 Output:

```
Running Buffer Management tests....

  Test 1 does a simple test of normal buffer manager operations:
  - Allocate a bunch of new pages
  - Write something on each one
  - Read that something back from each one
   (because we're buffering, this is where most of the writes happen)
  - Free the pages again
  Test 1 completed successfully.

  Test 2 exercises some illegal buffer manager operations:
  - Try to pin more pages than there are frames
*** Pinning too many pages
  --> Failed as expected

  - Try to free a doubly-pinned page
*** Freeing a pinned page
  --> Failed as expected

  - Try to unpin a page not in the buffer pool
*** Unpinning a page not in the buffer pool
  --> Failed as expected
  Test 2 completed successfully.

  Test 3 exercises some of the internals of the buffer manager
  - Allocate and dirty some new pages, one at a time, and leave some pinned
  - Read the pages
  Test 3 completed successfully.

...Buffer Management tests completely successfully.
```

Part 2 Implementation:
– The heapfile uses another class called heapfileP which uses the Tuple properties for delete record and insert record.
– Heap also uses a method to find whether a page exists by using 2 rids, 2 pageids, 2 hfpages.
– Scan uses another class called ScanP which has information about a given page.It also uses a method called next to find if there is another page after the next page

Part 2 Output:

```
Running Heap File tests....

Replacer: Clock


  Test 1: Insert and scan fixed-size records

  - Create a heap file

  - Add 100 records to the file

  - Scan the records just inserted

  Test 1 completed successfully.


  Test 2: Delete fixed-size records

  - Open the same heap file as test 1

  - Delete half the records

  - Scan the remaining records
  Test 2 completed successfully.


  Test 3: Update fixed-size records

  - Open the same heap file as tests 1 and 2

  - Change the records

  - Check that the updates are really there

  Test 3 completed successfully.


  Test 4: Test some error conditions

  - Try to change the size of a record

**** Shortening a record
  --> Failed as expected

**** Lengthening a record
  --> Failed as expected

  - Try to insert a record that's too long

**** Inserting a too-long record
  --> Failed as expected

  Test 4 completed successfully.


...Heap File tests completely successfully.
```