Lab 5: Max Heap

Implement a max-heap to manage print jobs. You are given the PrintJob class.

Class PrintJob

PrintJob.h

```
#ifndef __PRINTJOB_H
#define __PRINTJOB_H

using namespace std;

class PrintJob {
private:
  int priority;
  int jobNumber;
  int numPages;

public:
  PrintJob ( int, int, int);
  int getPriority ( );
  int getJobNumber ( );
  int getPages ( );
  //You can add additional functions here
};
#endif
```

PrintJob.cpp

```
#include "PrintJob.h"

PrintJob::PrintJob ( int setP, int setJ, int numP ):priority
(setP), jobNumber(setJ), numPages(numP){}
```

```
int PrintJob::getPriority ( ){
    return priority;
}

int PrintJob::getJobNumber ( ){
    return jobNumber;
}

int PrintJob::getPages ( ){
    return numPages;
}
```

class Heap

```
#ifndef __HEAP_H
#define __HEAP_H

#include "PrintJob.h"

const int MAX_HEAP_SIZE = 10;

class Heap {
private:
  PrintJob* arr[MAX_HEAP_SIZE];      // Notice this is an array of PrintJob pointers
  int numItems;  //current number of items in heap

public:
  /*Initializes an empty heap.*/
  Heap();

  /*Inserts a PrintJob to the heap without
  violating max-heap properties.*/
  void enqueue ( PrintJob* );
  /*Removes the node with highest priority from the heap.
  Follow the algorithm on priority-queue slides. */
  void dequeue ( );

  /*Returns the node with highest priority.*/
  PrintJob* highest ( );

  /*Prints the PrintJob with highest priority in the following format:
  Priority: priority, Job Number: jobNum, Number of Pages: numPages
  (Add a new line at the end.)*/
  void print ( );

private:
  /*This function is called by dequeue function
  to move the new root down the heap to the
  appropriare location.*/
  void trickleDown(int);

  //You can include additional private helper functions here
};
#endif
```

main.cpp

Use the following main function to test your program:

```
#include <iostream>
#include "Heap.h"

using namespace std;

int menu() {
  int choice = 0;
  cout << endl << "Enter menu choice: ";
  cout << endl;
  cout
    << "1. Enqueue" << endl
    << "2. Print" << endl
    << "3. Dequeue" << endl
    << "4. Quit" << endl;
  cin >> choice;

  // fix buffer just in case non-numeric choice entered
  // also gets rid of newline character
  cin.clear();
  cin.ignore(256, '\n');
  return choice;
}

int main(){
    Heap max_heap;

    int choice = menu();

    while (choice != 4) {

    if (choice == 1) {
      int priority, jobNumber, numPages;
      cout << "Enter print job to enqueue (priority, job Number, number of pages): ";
      cin>>priority>>jobNumber>>numPages;
      cout << endl;

      max_heap.enqueue(new PrintJob(priority, jobNumber, numPages));

    }
    else if (choice == 2) {
      max_heap.print();
    }
    else if (choice == 3) {
        max_heap.dequeue();
    }
    //fix buffer just in case non-numeric choice entered
    choice = menu();
    }
    return 0;
}
```