Project #3

Due Dates: Saturday, October 29 at 11:59pm

Submit: eLearning

Late Policy: 24-hour late period, then zero

Instructions: This is an individual assignment. Answers should be your own work.

Introduction:

In this project you will apply the heap data structure to a problem.

Description:

The problem is to simulate the operation of a printer that prints the

jobs according to priority.

The priority of a job is determined by two factors:

user\_priority: an integer from 1 to 3 (1 is highest)

numpages: pages to be printed

Job priority = user\_priority \* numpages

For example, consider two users:

Joe: user\_priority = 3, numpages=50

Sue: user\_priority = 1, numpages=10

Joe's priority = 3x50 = 150

Sue's priority = 1x10 = 10

Since Sue has a high user-priority and is only printing 10 pages, she will

have priority over Joe.

A job to print should be represented by a class named Printjob. This class

should contain the user's name, the user's priority, and number of pages. It

should implement Comparable with compareTo based on job priority.

Derive a subclass of Printjob called OutsidePrintjob. These are just like

Printjobs, but they compute a cost based on 10 cents per page.

Another class called Printer should read an input file and create objects

for each entry. These objects should be added to a priority queue

using the textbook's BinaryHeap class (unmodified).

The input file contains each job to print on a separate line, with tabs between

the fields. The fields are name, user priority, pages, and a flag indicating

inside or outside job (I or O).

Once the file is read and the print jobs have been added to the binary heap,

the Printer object should deleteMin each job and print its user's name, user priority,

and pages to the screen. OutsidePrintjobs should also show their cost.

Submit to eLearning:

Printjob.java

OutsidePrintjob.java

Printer.java

int temp=p1.deleteMin();

for(int j=1;j<4;j++)

{

for(int i = 0; i < arr1.size(); i++)

{

if(arr1.get(i).jobPriority==temp && arr1.get(i).priority==j)

{

System.out.println(arr1.get(i));

arr1.remove(i);

}

}

}

ArrayList<Printjob> arr1=new ArrayList<>();

arr1.add(temp);

arr1.add(temp);