

Sabanci University

Faculty of Engineering and Natural Sciences

CS204 Advanced Programming

Summer 2022-2023

Take-Home Exam 4

Printer System Simulation
using Operator Overloading and Multithreading Techniques

Due: 25 August 2023 11.55pm (SHARP)

DISCLAIMER:

Only checking the sample run cases might not be sufficient as your solution will be checked against a variety of samples different from the provided samples; however checking and studying these cases is highly encouraged and recommended.

You can NOT collaborate with your friends and discuss your solutions with each other. You have to write down the code on your own. Plagiarism will not be tolerated AND cooperation is not an excuse!

Introduction

In this take-home exam (THE), you are asked to write a multithreaded C++ program that simulates a printer queue where print jobs are being sent to the queue by multiple users. There are 3 users sharing the queue of a single printer, as depicted in Figure 1. In this THE, you will simulate the process of users generating print jobs and sending them to the print queue, and the printer processing those print jobs by getting them out from the queue. The printer queue is to be implemented in terms of the dynamic queue data structure. As an incomplete implementation is provided in this THE package, one of your tasks is to complete this implementation of the dynamic queue class such that it will overload four operators (`=`, `-`, `+`, `<<`), as well as having proper implementations for a copy constructor and/or a destructor, if necessary. Your other task is to implement the simulation mentioned above using multithreading techniques. Needless to say, your main program should make use of the operator overloading functions mentioned above.

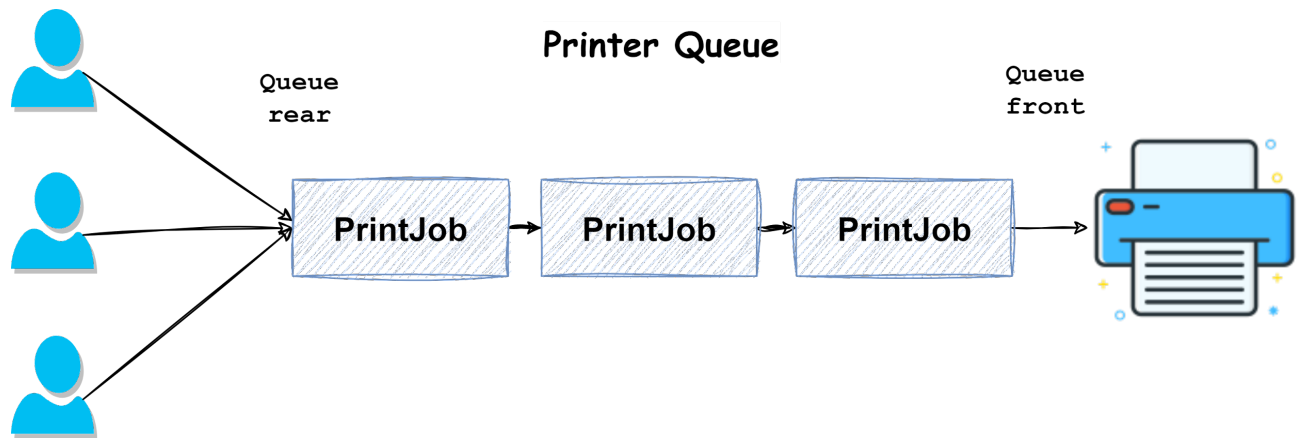



Figure 1: three users, printer queue and the printer

When a user () sends a print job to the queue, s/he will wait sometime after sending this print job. This waiting time (after a user sends a print job to the queue) should be probabilistic. Thus, the minimum and maximum range values for this duration will be a user input. Moreover, the number of pages in each print job is also probabilistic, and the minimum/maximum range values of it will also be a user input read from the keyboard at the very beginning of the program (See Section "Details of the Simulation" for details).

In the scope of this THE, simulation means to employ a printer queue where print jobs, with a random amount of pages, are being sent to the rear of the queue at random intervals, and the printer processes those print jobs by getting them from the front of the queue one by one. Simulation starts after taking the inputs from the keyboard and continues until all of the print jobs are processed (printed). During the simulation, your program will display some verbose output about the actions of the users, the contents of the printer queue, and the printing task itself (see Section "Details of the Simulation" for details and the Sample Runs for the output format).

Using Threads

There should be four (4) threads (other than the main thread) in your program. Three (3) of them are for the users, one for each; and the other one is for the printer. In the user threads, printing tasks with a random number of pages will first be generated, and then be enqueued to the rear of the queue. On the other hand, in the printer thread, these print jobs will be taken out from the front of the queue (i.e., dequeue) and processed one at a time (See Section "Using Operator Overloading" for information about the enqueue process through the **+** operator, and the dequeue process through the **-** operator). There should be only one queue

shared by all the threads. Since there are three (3) users, print jobs will be enqueued to the same queue during the simulation by different users simultaneously. However, there is only one printer dequeuing and printing the jobs from that queue. This indeed is one of the very widely known producer consumer problem examples.

Using Operator Overloading

In this THE, you will overload four operators, namely +, -, = and <<, to work properly and consistently with the provided `PrintQueue` class.

- The `+` operator will work as enqueue to the queue, where it will receive a print job as its operand that will be enqueued to the caller queue object. The overloading should return a version of the queue with the new item inserted, rather than performing the required operation inline.
- The `-` operator will work as dequeue to the queue, where it will receive a print job as its operand as a reference variable to store the information of the dequeued print job. The overloading should return a version of the queue with the given item removed, rather than performing the required operation inline.
- The `=` operator will be used along with the `+` and `-` operators to assign an instance from the `PrintQueue` class into another instance of it after adding (i.e., enqueueing) or subtracting (i.e., dequeuing) a print job to/from it.
- The `<<` operator will be used to output the contents of the print queue.

Please note that, if an overloading can be handled via a member function, the implementation should be carried out that way.

Usage example

Below is a few lines of code to show a possible usage of the above operators in the program according to the provided `PrintQueue` class and the `PrintJobNode` struct:

```
PrintQueue printQ;
.....
printQ = printQ + PrintJobNode(.....); // enqueue a job to the rear of the printer queue
.....
PrintJobNode JN;
printQ = printQ - JN; // dequeue a job from the front of the printer queue
.....
cout << printQ << endl; // display the content of the printer queue on the screen
```

Details of the Simulation

Before the simulation begins, a total of 5 (five) inputs will be entered via the keyboard. The first one is the *maxPrintJobs*; maximum number of print jobs to be generated in the program (not per user, total for all users). In other words, the simulation will finish when *maxPrintJobs* have been printed by the printer. Moreover, your program should not allow more than *maxPrintJobs* to be generated in total. Thus, at the end of the simulation, all of the print jobs should be processed by the printer and the queue must be empty.

Second and third input values are the minimum and maximum boundaries for the random time duration (in seconds) that a user should wait after creating a print job. Those values will determine the range of the random number that will be generated to allow a user thread to sleep. Moreover, each user thread should sleep for a random amount of time before creating the first print job.

Fourth and fifth input values are the minimum and maximum boundaries for the range of the random number of pages that will be generated for each print job. Assuming that one page takes one second to print, you have to sleep the printer thread accordingly to simulate the printing operation.

To generate the random numbers mentioned above for the time duration and for the number of pages, please **do not use** the RandGen class from CS201 since it is not thread-safe in multithreaded applications. Instead, **you must use the following thread-safe function, `random_range`, for generating random number of pages and waiting times inside the threads.** This function returns a random number between (and including) its `min` and `max` parameters. In the threads, you can call it by passing the minimum and maximum boundaries of the corresponding thread as arguments.

```
#include <random>
#include <time.h>

int random_range(const int & min, const int & max) {
    static mt19937 generator(time(0));
    uniform_int_distribution<int> distribution(min, max);
    return distribution(generator);
}
```


Simulation starts when all the inputs are entered through the keyboard. At the beginning of the program (after getting the inputs and just before the beginning of the simulation), your program has to display a message saying that the simulation is starting and display the current time.

Note about the displaying the current time in Windows vs in MacOS:

Please check the lecture slides on how to get the current time.

For Windows users, you can use the code snippet from the slides ;

For MacOS users, you will need to change `localtime_s(ptm, &tt);` to `ptm = localtime(&tt);`

During the simulation, users () will generate print jobs in their corresponding threads as explained before. Each new print job must be associated with a consecutive ID number starting with 1. You may simulate the waiting time interval (that the user will wait) after each print job (and before creating the first print job) by sleeping the threads using the `thread` and `chrono` libraries and the `this_thread::sleep_for(chrono::seconds(time_in_seconds))` command.

During the simulation, each user thread should display information about the print job that is sent to the queue. Basically, it will display a message that includes the user ID, print job ID, the number of pages of that print job, the size of the queue after enqueueing, and the time of sending the print job.

Also, the printer thread should display the details of the processed print job. Basically, it will display a message that includes the print job ID, its number of pages and the queue size after the job has been taken from the queue, and the time of start of printing. Then, it will display the content of the queue (again, after the job has been taken from the queue). If the queue then becomes empty, it should print a specific message indicating that.

You have to display another message at the end of printing as well, using the same data about the print job except the queue size. Please note that one page of printing takes one second, and do not forget to sleep each of the user threads before creating the first print job.

At the end of the simulation, a message saying that the simulation has ended must be displayed together with the current time. Please note that since you sleep the user threads after creating a print job, there could be some time gaps between the end of processing of the last print job and the end of simulation; this basically depends on the min and max parameters, and the random values generated.

Please see the Sample Runs section for some examples and specific messages format.

Two important rules about the simulation details and use of threads:

- Note that a single line of output from a particular thread may be interleaved by the output of another thread, if you do not take appropriate precautions. If this happens, the outputs mix up and become messy. You have to code appropriately, with the help of a dedicated mutex, in order not to end up with such an undesirable situation. Here by "dedicated" we mean; the mutex that you use for tidy output should not be the one that you use for accessing the shared queue object. These two critical sections are different concepts and must be handled via different mutexes.
- *Do not sleep a thread while a mutex is locked.* We use sleeping to simulate the waiting duration and printing duration (as number of pages). We do not access shared resources while doing so. Thus, sleeping a thread while a mutex is locked is totally nonsense.

Any violation to these rules will be penalized severely (50 -or even more-points).

PrintQueue class and use of global variables

We provide a header file (`PrintQueue.h`) and a partial implementation (`PrintQueue.cpp`) together with this THE. You need to update both the header file and the cpp file with the declaration and implementation of the four overloaded operators that were mentioned above, and any other required functionality, as well as a destructor and a copy constructor, if necessary. Also, you have to use these operators in your main program to demonstrate their usage.

Finally, good news; you may use global variables in this THE. Actually, it would be miserable not to use them in a program that has several threads. However, we kindly request you not to exaggerate the global usage since after a certain point you may lose control over your program (as the famous Turkish proverb says "azı karar, çoğu zarar").

Some versions of C++ compilers impose a limitation for the number of parameters on the thread constructor. It is 5 (except the entry point function address) for

VC++ 2012 and if you exceed this, you receive compilation errors. If this amount of parameters is not sufficient for your algorithm, you may make some of them global, or pack them into structs/arrays. There is also a way to increase this limit using the steps explained below (stackoverflow.com); however, since you cannot know whether the grader has the same setup or not, it is better to stay within the standard limit

Right click on your project in Solution Explorer. Select Properties... and navigate to C/C++ -> Preprocessor and set add _VARIADIC_MAX=10 to "Preprocessor Definitions". Make sure you do that for all the configurations of the project (debug, release, etc.) – JP Flouret Nov 13 '14 at 22:44

Sample Runs

Some sample runs are given below, but these are not comprehensive, therefore you have to consider all cases, to get a full mark.

Due to the probabilistic nature of the THE and due to scheduling of threads, same inputs may yield different outputs for your code. However, the order of the events must be consistent with the THE requirements and the given inputs. Nevertheless, occasional (i.e., rare) inconsistencies in the display order of the events occurring at the same time are acceptable.

The inputs from the keyboard are written in **bold**

Sample Run 1:

```
Please enter the max number of print jobs: 1
Please enter the min and max values for the time range between two print jobs:
Min: 1
Max: 1
Please enter the min and max values for the number of pages in a print job:
Min number of pages: 1
Max number of pages: 1
Simulation starts 15:45:51
User 2 sent new print job with ID 1 sent to the printer queue, number of pages: 1 (print queue size: 1) 15:45:52
The printer started to print the job with ID: 1, number of pages: 1 (queue size is: 0) 15:45:52
The print queue is empty.

The printer finished printing the job with ID: 1, number of pages: 1 15:45:53
End of the simulation at: 15:45:53
```

Sample Run 2:

```
Please enter the max number of print jobs: 1
Please enter the min and max values for the time range between two print jobs:
Min: 1
```

Max: **1**

Please enter the min and max values for the number of pages in a print job:

Min number of pages: **1**

Max number of pages: **5**

Simulation starts 15:59:28

User 2 sent new print job with ID 1 sent to the printer queue, number of pages: 4 (print queue size: 1) 15:59:29

The printer started to print the job with ID: 1, number of pages: 4 (queue size is: 0) 15:59:29

The print queue is empty.

The printer finished printing the job with ID: 1, number of pages: 4 15:59:33

End of the simulation at: 15:59:33

Sample Run 3:

Please enter the max number of print jobs: **3**

Please enter the min and max values for the time range between two print jobs:

Min: **1**

Max: **1**

Please enter the min and max values for the number of pages in a print job:

Min number of pages: **1**

Max number of pages: **1**

Simulation starts 15:02:34

User 3 sent new print job with ID 1 sent to the printer queue, number of pages: 1 (print queue size: 1) 15:02:35

User 2 sent new print job with ID 2 sent to the printer queue, number of pages: 1 (print queue size: 2) 15:02:35

User 1 sent new print job with ID 3 sent to the printer queue, number of pages: 1 (print queue size: 3) 15:02:35

The printer started to print the job with ID: 1, number of pages: 1 (queue size is: 2) 15:02:35

The print queue contains:

Print Job ID: 2, Pages: 1

Print Job ID: 3, Pages: 1

The printer finished printing the job with ID: 1, number of pages: 1 15:02:36

The printer started to print the job with ID: 2, number of pages: 1 (queue size is: 1) 15:02:36

The print queue contains:

Print Job ID: 3, Pages: 1

The printer finished printing the job with ID: 2, number of pages: 1 15:02:37

The printer started to print the job with ID: 3, number of pages: 1 (queue size is: 0) 15:02:37

The print queue is empty.

The printer finished printing the job with ID: 3, number of pages: 1 15:02:38

End of the simulation at: 15:02:38

Sample Run 4:

Please enter the max number of print jobs: **3**

Please enter the min and max values for the time range between two print jobs:

Min: **1**

Max: **2**

Please enter the min and max values for the number of pages in a print job:

Min number of pages: **1**

Max number of pages: **3**

Simulation starts 15:52:16

User 1 sent new print job with ID 1 sent to the printer queue, number of pages: 3 (print queue size: 1) 15:52:17

The printer started to print the job with ID: 1, number of pages: 3 (queue size is: 0) 15:52:17

The print queue is empty.

User 2 sent new print job with ID 2 sent to the printer queue, number of pages: 3 (print queue size: 1) 15:52:17

User 3 sent new print job with ID 3 sent to the printer queue, number of pages: 2 (print queue size: 2) 15:52:18

The printer finished printing the job with ID: 1, number of pages: 3 15:52:20

The printer started to print the job with ID: 2, number of pages: 3 (queue size is: 1) 15:52:20

The print queue contains:

Print Job ID: 3, Pages: 2

The printer finished printing the job with ID: 2, number of pages: 3 15:52:23

The printer started to print the job with ID: 3, number of pages: 2 (queue size is: 0) 15:52:23

The print queue is empty.

The printer finished printing the job with ID: 3, number of pages: 2 15:52:25

End of the simulation at: 15:52:25

Sample Run 5:

Please enter the max number of print jobs: **9**

Please enter the min and max values for the time range between two print jobs:

Min: **1**

Max: **2**

Please enter the min and max values for the number of pages in a print job:

Min number of pages: **2**

Max number of pages: **7**

Simulation starts 15:53:57

User 2 sent new print job with ID 1 sent to the printer queue, number of pages: 7 (print queue size: 1) 15:53:58

The printer started to print the job with ID: 1, number of pages: 7 (queue size is: 0) 15:53:58

The print queue is empty.

User 3 sent new print job with ID 2 sent to the printer queue, number of pages: 5 (print queue size: 1) 15:53:59

User 1 sent new print job with ID 3 sent to the printer queue, number of pages: 2 (print queue size: 2) 15:53:59

User 2 sent new print job with ID 4 sent to the printer queue, number of pages: 7 (print queue size: 3) 15:54:00

User 3 sent new print job with ID 5 sent to the printer queue, number of pages: 3 (print queue size: 4) 15:54:00

User 2 sent new print job with ID 6 sent to the printer queue, number of pages: 7 (print queue size: 5) 15:54:01

User 1 sent new print job with ID 7 sent to the printer queue, number of pages: 2 (print queue size: 6) 15:54:01

User 3 sent new print job with ID 8 sent to the printer queue, number of pages: 4 (print queue size: 7) 15:54:01

User 2 sent new print job with ID 9 sent to the printer queue, number of pages: 2 (print queue size: 8) 15:54:02

The printer finished printing the job with ID: 1, number of pages: 7 15:54:05

The printer started to print the job with ID: 2, number of pages: 5 (queue size is: 7) 15:54:05

The print queue contains:

Print Job ID: 3, Pages: 2

Print Job ID: 4, Pages: 7

Print Job ID: 5, Pages: 3

Print Job ID: 6, Pages: 7

Print Job ID: 7, Pages: 2

Print Job ID: 8, Pages: 4

Print Job ID: 9, Pages: 2

The printer finished printing the job with ID: 2, number of pages: 5 15:54:10

The printer started to print the job with ID: 3, number of pages: 2 (queue size is: 6) 15:54:10

The print queue contains:

Print Job ID: 4, Pages: 7

Print Job ID: 5, Pages: 3

Print Job ID: 6, Pages: 7

Print Job ID: 7, Pages: 2

Print Job ID: 8, Pages: 4
Print Job ID: 9, Pages: 2

The printer finished printing the job with ID: 3, number of pages: 2 15:54:12
The printer started to print the job with ID: 4, number of pages: 7 (queue size is: 5) 15:54:12
The print queue contains:
Print Job ID: 5, Pages: 3
Print Job ID: 6, Pages: 7
Print Job ID: 7, Pages: 2
Print Job ID: 8, Pages: 4
Print Job ID: 9, Pages: 2

The printer finished printing the job with ID: 4, number of pages: 7 15:54:19
The printer started to print the job with ID: 5, number of pages: 3 (queue size is: 4) 15:54:19
The print queue contains:
Print Job ID: 6, Pages: 7
Print Job ID: 7, Pages: 2
Print Job ID: 8, Pages: 4
Print Job ID: 9, Pages: 2

The printer finished printing the job with ID: 5, number of pages: 3 15:54:22
The printer started to print the job with ID: 6, number of pages: 7 (queue size is: 3) 15:54:22
The print queue contains:
Print Job ID: 7, Pages: 2
Print Job ID: 8, Pages: 4
Print Job ID: 9, Pages: 2

The printer finished printing the job with ID: 6, number of pages: 7 15:54:29
The printer started to print the job with ID: 7, number of pages: 2 (queue size is: 2) 15:54:29
The print queue contains:
Print Job ID: 8, Pages: 4
Print Job ID: 9, Pages: 2

The printer finished printing the job with ID: 7, number of pages: 2 15:54:31
The printer started to print the job with ID: 8, number of pages: 4 (queue size is: 1) 15:54:31
The print queue contains:
Print Job ID: 9, Pages: 2

The printer finished printing the job with ID: 8, number of pages: 4 15:54:35
The printer started to print the job with ID: 9, number of pages: 2 (queue size is: 0) 15:54:35
The print queue is empty.

The printer finished printing the job with ID: 9, number of pages: 2 15:54:37
End of the simulation at: 15:54:37

Sample Run 6:

Please enter the max number of print jobs: **12**
Please enter the min and max values for the time range between two print jobs:
Min: **1**
Max: **1**
Please enter the min and max values for the number of pages in a print job:
Min number of pages: **3**
Max number of pages: **7**
Simulation starts 19:07:20
User 3 sent new print job with ID 1 sent to the printer queue, number of pages: 3 (print queue size: 1) 19:07:21

The printer started to print the job with ID: 1, number of pages: 3 (queue size is: 0) 19:07:21
The print queue is empty.

User 2 sent new print job with ID 2 sent to the printer queue, number of pages: 7 (print queue size: 1) 19:07:21
User 1 sent new print job with ID 3 sent to the printer queue, number of pages: 5 (print queue size: 2) 19:07:21
User 1 sent new print job with ID 4 sent to the printer queue, number of pages: 5 (print queue size: 3) 19:07:22
User 2 sent new print job with ID 5 sent to the printer queue, number of pages: 6 (print queue size: 4) 19:07:22
User 3 sent new print job with ID 6 sent to the printer queue, number of pages: 3 (print queue size: 5) 19:07:22
User 2 sent new print job with ID 7 sent to the printer queue, number of pages: 7 (print queue size: 6) 19:07:23
User 1 sent new print job with ID 8 sent to the printer queue, number of pages: 3 (print queue size: 7) 19:07:23
User 3 sent new print job with ID 9 sent to the printer queue, number of pages: 4 (print queue size: 8) 19:07:23
The printer finished printing the job with ID: 1, number of pages: 3 19:07:24

The printer started to print the job with ID: 2, number of pages: 7 (queue size is: 7) 19:07:24

The print queue contains:

Print Job ID: 3, Pages: 5
Print Job ID: 4, Pages: 5
Print Job ID: 5, Pages: 6
Print Job ID: 6, Pages: 3
Print Job ID: 7, Pages: 7
Print Job ID: 8, Pages: 3
Print Job ID: 9, Pages: 4

User 3 sent new print job with ID 10 sent to the printer queue, number of pages: 4 (print queue size: 8) 19:07:24
User 1 sent new print job with ID 11 sent to the printer queue, number of pages: 6 (print queue size: 9) 19:07:24
User 2 sent new print job with ID 12 sent to the printer queue, number of pages: 7 (print queue size: 10)
19:07:24

The printer finished printing the job with ID: 2, number of pages: 7 19:07:31

The printer started to print the job with ID: 3, number of pages: 5 (queue size is: 9) 19:07:31

The print queue contains:

Print Job ID: 4, Pages: 5
Print Job ID: 5, Pages: 6
Print Job ID: 6, Pages: 3
Print Job ID: 7, Pages: 7
Print Job ID: 8, Pages: 3
Print Job ID: 9, Pages: 4
Print Job ID: 10, Pages: 4
Print Job ID: 11, Pages: 6
Print Job ID: 12, Pages: 7

The printer finished printing the job with ID: 3, number of pages: 5 19:07:36

The printer started to print the job with ID: 4, number of pages: 5 (queue size is: 8) 19:07:36

The print queue contains:

Print Job ID: 5, Pages: 6
Print Job ID: 6, Pages: 3
Print Job ID: 7, Pages: 7
Print Job ID: 8, Pages: 3
Print Job ID: 9, Pages: 4
Print Job ID: 10, Pages: 4
Print Job ID: 11, Pages: 6
Print Job ID: 12, Pages: 7

The printer finished printing the job with ID: 4, number of pages: 5 19:07:41

The printer started to print the job with ID: 5, number of pages: 6 (queue size is: 7) 19:07:41

The print queue contains:

Print Job ID: 6, Pages: 3
Print Job ID: 7, Pages: 7
Print Job ID: 8, Pages: 3
Print Job ID: 9, Pages: 4

Print Job ID: 10, Pages: 4
Print Job ID: 11, Pages: 6
Print Job ID: 12, Pages: 7

The printer finished printing the job with ID: 5, number of pages: 6 19:07:47
The printer started to print the job with ID: 6, number of pages: 3 (queue size is: 6) 19:07:47
The print queue contains:
Print Job ID: 7, Pages: 7
Print Job ID: 8, Pages: 3
Print Job ID: 9, Pages: 4
Print Job ID: 10, Pages: 4
Print Job ID: 11, Pages: 6
Print Job ID: 12, Pages: 7

The printer finished printing the job with ID: 6, number of pages: 3 19:07:50
The printer started to print the job with ID: 7, number of pages: 7 (queue size is: 5) 19:07:50
The print queue contains:
Print Job ID: 8, Pages: 3
Print Job ID: 9, Pages: 4
Print Job ID: 10, Pages: 4
Print Job ID: 11, Pages: 6
Print Job ID: 12, Pages: 7

The printer finished printing the job with ID: 7, number of pages: 7 19:07:57
The printer started to print the job with ID: 8, number of pages: 3 (queue size is: 4) 19:07:57
The print queue contains:
Print Job ID: 9, Pages: 4
Print Job ID: 10, Pages: 4
Print Job ID: 11, Pages: 6
Print Job ID: 12, Pages: 7

The printer finished printing the job with ID: 8, number of pages: 3 19:08:00
The printer started to print the job with ID: 9, number of pages: 4 (queue size is: 3) 19:08:00
The print queue contains:
Print Job ID: 10, Pages: 4
Print Job ID: 11, Pages: 6
Print Job ID: 12, Pages: 7

The printer finished printing the job with ID: 9, number of pages: 4 19:08:04
The printer started to print the job with ID: 10, number of pages: 4 (queue size is: 2) 19:08:04
The print queue contains:
Print Job ID: 11, Pages: 6
Print Job ID: 12, Pages: 7

The printer finished printing the job with ID: 10, number of pages: 4 19:08:08
The printer started to print the job with ID: 11, number of pages: 6 (queue size is: 1) 19:08:08
The print queue contains:
Print Job ID: 12, Pages: 7

The printer finished printing the job with ID: 11, number of pages: 6 19:08:14
The printer started to print the job with ID: 12, number of pages: 7 (queue size is: 0) 19:08:14
The print queue is empty.

The printer finished printing the job with ID: 12, number of pages: 7 19:08:21
End of the simulation at: 19:08:21

Sample Run 7:

Please enter the max number of print jobs: **12**

Please enter the min and max values for the time range between two print jobs:

Min: **2**

Max: **7**

Please enter the min and max values for the number of pages in a print job:

Min number of pages: **1**

Max number of pages: **1**

Simulation starts 19:02:36

User 1 sent new print job with ID 1 sent to the printer queue, number of pages: 1 (print queue size: 1) 19:02:41

User 2 sent new print job with ID 2 sent to the printer queue, number of pages: 1 (print queue size: 2) 19:02:41

The printer started to print the job with ID: 1, number of pages: 1 (queue size is: 1) 19:02:41

The print queue contains:

Print Job ID: 2, Pages: 1

The printer finished printing the job with ID: 1, number of pages: 1 19:02:42

The printer started to print the job with ID: 2, number of pages: 1 (queue size is: 0) 19:02:42

The print queue is empty.

User 3 sent new print job with ID 3 sent to the printer queue, number of pages: 1 (print queue size: 1) 19:02:43

The printer finished printing the job with ID: 2, number of pages: 1 19:02:43

The printer started to print the job with ID: 3, number of pages: 1 (queue size is: 1) 19:02:43

The print queue contains:

Print Job ID: 4, Pages: 1

User 1 sent new print job with ID 4 sent to the printer queue, number of pages: 1 (print queue size: 2) 19:02:43

The printer finished printing the job with ID: 3, number of pages: 1 19:02:44

The printer started to print the job with ID: 4, number of pages: 1 (queue size is: 0) 19:02:44

The print queue is empty.

The printer finished printing the job with ID: 4, number of pages: 1 19:02:45

User 3 sent new print job with ID 5 sent to the printer queue, number of pages: 1 (print queue size: 1) 19:02:46

User 2 sent new print job with ID 6 sent to the printer queue, number of pages: 1 (print queue size: 2) 19:02:46

The printer started to print the job with ID: 5, number of pages: 1 (queue size is: 1) 19:02:46

The print queue contains:

Print Job ID: 6, Pages: 1

The printer finished printing the job with ID: 5, number of pages: 1 19:02:47

The printer started to print the job with ID: 6, number of pages: 1 (queue size is: 1) 19:02:47

The print queue contains:

Print Job ID: 7, Pages: 1

User 1 sent new print job with ID 7 sent to the printer queue, number of pages: 1 (print queue size: 2) 19:02:47

The printer finished printing the job with ID: 6, number of pages: 1 19:02:48

The printer started to print the job with ID: 7, number of pages: 1 (queue size is: 0) 19:02:48

The print queue is empty.

User 2 sent new print job with ID 8 sent to the printer queue, number of pages: 1 (print queue size: 1) 19:02:49

The printer finished printing the job with ID: 7, number of pages: 1 19:02:49

The printer started to print the job with ID: 8, number of pages: 1 (queue size is: 0) 19:02:49

The print queue is empty.

User 3 sent new print job with ID 9 sent to the printer queue, number of pages: 1 (print queue size: 1) 19:02:50

The printer finished printing the job with ID: 8, number of pages: 1 19:02:50

The printer started to print the job with ID: 9, number of pages: 1 (queue size is: 0) 19:02:50

The print queue is empty.

The printer finished printing the job with ID: 9, number of pages: 1 19:02:51

User 1 sent new print job with ID 10 sent to the printer queue, number of pages: 1 (print queue size: 1) 19:02:52
The printer started to print the job with ID: 10, number of pages: 1 (queue size is: 0) 19:02:52
The print queue is empty.

User 3 sent new print job with ID 11 sent to the printer queue, number of pages: 1 (print queue size: 1) 19:02:53
The printer finished printing the job with ID: 10, number of pages: 1 19:02:53
The printer started to print the job with ID: 11, number of pages: 1 (queue size is: 0) 19:02:53
The print queue is empty.

User 1 sent new print job with ID 12 sent to the printer queue, number of pages: 1 (print queue size: 1) 19:02:54
The printer finished printing the job with ID: 11, number of pages: 1 19:02:54
The printer started to print the job with ID: 12, number of pages: 1 (queue size is: 0) 19:02:54
The print queue is empty.

The printer finished printing the job with ID: 12, number of pages: 1 19:02:55
End of the simulation at: 19:03:01

Sample Run 8:

Please enter the max number of print jobs: **11**
Please enter the min and max values for the time range between two print jobs:
Min: **1**
Max: **8**
Please enter the min and max values for the number of pages in a print job:
Min number of pages: **3**
Max number of pages: **11**
Simulation starts 18:58:34

User 1 sent new print job with ID 1 sent to the printer queue, number of pages: 10 (print queue size: 1) 18:58:35
The printer started to print the job with ID: 1, number of pages: 10 (queue size is: 0) 18:58:35
The print queue is empty.

User 2 sent new print job with ID 2 sent to the printer queue, number of pages: 11 (print queue size: 1) 18:58:37
User 1 sent new print job with ID 3 sent to the printer queue, number of pages: 8 (print queue size: 2) 18:58:38
User 3 sent new print job with ID 4 sent to the printer queue, number of pages: 5 (print queue size: 3) 18:58:39
User 1 sent new print job with ID 5 sent to the printer queue, number of pages: 11 (print queue size: 4) 18:58:42
User 2 sent new print job with ID 6 sent to the printer queue, number of pages: 7 (print queue size: 5) 18:58:44
User 3 sent new print job with ID 7 sent to the printer queue, number of pages: 3 (print queue size: 6) 18:58:44
The printer finished printing the job with ID: 1, number of pages: 10 18:58:45
The printer started to print the job with ID: 2, number of pages: 11 (queue size is: 5) 18:58:45
The print queue contains:
Print Job ID: 3, Pages: 8
Print Job ID: 4, Pages: 5
Print Job ID: 5, Pages: 11
Print Job ID: 6, Pages: 7
Print Job ID: 7, Pages: 3

User 1 sent new print job with ID 8 sent to the printer queue, number of pages: 8 (print queue size: 6) 18:58:45
User 3 sent new print job with ID 9 sent to the printer queue, number of pages: 7 (print queue size: 7) 18:58:50
User 3 sent new print job with ID 10 sent to the printer queue, number of pages: 5 (print queue size: 8) 18:58:51
User 1 sent new print job with ID 11 sent to the printer queue, number of pages: 8 (print queue size: 9) 18:58:51
The printer finished printing the job with ID: 2, number of pages: 11 18:58:56
The printer started to print the job with ID: 3, number of pages: 8 (queue size is: 8) 18:58:56
The print queue contains:
Print Job ID: 4, Pages: 5
Print Job ID: 5, Pages: 11

Print Job ID: 6, Pages: 7
Print Job ID: 7, Pages: 3
Print Job ID: 8, Pages: 8
Print Job ID: 9, Pages: 7
Print Job ID: 10, Pages: 5
Print Job ID: 11, Pages: 8

The printer finished printing the job with ID: 3, number of pages: 8 18:59:04
The printer started to print the job with ID: 4, number of pages: 5 (queue size is: 7) 18:59:04
The print queue contains:
Print Job ID: 5, Pages: 11
Print Job ID: 6, Pages: 7
Print Job ID: 7, Pages: 3
Print Job ID: 8, Pages: 8
Print Job ID: 9, Pages: 7
Print Job ID: 10, Pages: 5
Print Job ID: 11, Pages: 8

The printer finished printing the job with ID: 4, number of pages: 5 18:59:09
The printer started to print the job with ID: 5, number of pages: 11 (queue size is: 6) 18:59:09
The print queue contains:
Print Job ID: 6, Pages: 7
Print Job ID: 7, Pages: 3
Print Job ID: 8, Pages: 8
Print Job ID: 9, Pages: 7
Print Job ID: 10, Pages: 5
Print Job ID: 11, Pages: 8

The printer finished printing the job with ID: 5, number of pages: 11 18:59:20
The printer started to print the job with ID: 6, number of pages: 7 (queue size is: 5) 18:59:20
The print queue contains:
Print Job ID: 7, Pages: 3
Print Job ID: 8, Pages: 8
Print Job ID: 9, Pages: 7
Print Job ID: 10, Pages: 5
Print Job ID: 11, Pages: 8

The printer finished printing the job with ID: 6, number of pages: 7 18:59:28
The printer started to print the job with ID: 7, number of pages: 3 (queue size is: 4) 18:59:28
The print queue contains:
Print Job ID: 8, Pages: 8
Print Job ID: 9, Pages: 7
Print Job ID: 10, Pages: 5
Print Job ID: 11, Pages: 8

The printer finished printing the job with ID: 7, number of pages: 3 18:59:31
The printer started to print the job with ID: 8, number of pages: 8 (queue size is: 3) 18:59:31
The print queue contains:
Print Job ID: 9, Pages: 7
Print Job ID: 10, Pages: 5
Print Job ID: 11, Pages: 8

The printer finished printing the job with ID: 8, number of pages: 8 18:59:39
The printer started to print the job with ID: 9, number of pages: 7 (queue size is: 2) 18:59:39
The print queue contains:
Print Job ID: 10, Pages: 5
Print Job ID: 11, Pages: 8

The printer finished printing the job with ID: 9, number of pages: 7 18:59:46
The printer started to print the job with ID: 10, number of pages: 5 (queue size is: 1) 18:59:46
The print queue contains:
Print Job ID: 11, Pages: 8

The printer finished printing the job with ID: 10, number of pages: 5 18:59:51
The printer started to print the job with ID: 11, number of pages: 8 (queue size is: 0) 18:59:51
The print queue is empty.

The printer finished printing the job with ID: 11, number of pages: 8 18:59:59
End of the simulation at: 18:59:59

Sample Run 9:

Please enter the max number of print jobs: **6**
Please enter the min and max values for the time range between two print jobs:
Min: **1**
Max: **5**
Please enter the min and max values for the number of pages in a print job:
Min number of pages: **2**
Max number of pages: **5**
Simulation starts 19:09:25
User 3 sent new print job with ID 1 sent to the printer queue, number of pages: 5 (print queue size: 1) 19:09:27
User 1 sent new print job with ID 2 sent to the printer queue, number of pages: 3 (print queue size: 2) 19:09:27
User 2 sent new print job with ID 3 sent to the printer queue, number of pages: 3 (print queue size: 3) 19:09:27
The printer started to print the job with ID: 1, number of pages: 5 (queue size is: 2) 19:09:27
The print queue contains:
Print Job ID: 2, Pages: 3
Print Job ID: 3, Pages: 3

User 1 sent new print job with ID 4 sent to the printer queue, number of pages: 3 (print queue size: 3) 19:09:28
User 3 sent new print job with ID 5 sent to the printer queue, number of pages: 5 (print queue size: 4) 19:09:28
User 2 sent new print job with ID 6 sent to the printer queue, number of pages: 3 (print queue size: 5) 19:09:28
The printer finished printing the job with ID: 1, number of pages: 5 19:09:32
The printer started to print the job with ID: 2, number of pages: 3 (queue size is: 4) 19:09:32
The print queue contains:
Print Job ID: 3, Pages: 3
Print Job ID: 4, Pages: 3
Print Job ID: 5, Pages: 5
Print Job ID: 6, Pages: 3

The printer finished printing the job with ID: 2, number of pages: 3 19:09:35
The printer started to print the job with ID: 3, number of pages: 3 (queue size is: 3) 19:09:35
The print queue contains:
Print Job ID: 4, Pages: 3
Print Job ID: 5, Pages: 5
Print Job ID: 6, Pages: 3

The printer finished printing the job with ID: 3, number of pages: 3 19:09:38
The printer started to print the job with ID: 4, number of pages: 3 (queue size is: 2) 19:09:38
The print queue contains:
Print Job ID: 5, Pages: 5
Print Job ID: 6, Pages: 3

The printer finished printing the job with ID: 4, number of pages: 3 19:09:41

The printer started to print the job with ID: 5, number of pages: 5 (queue size is: 1) 19:09:41
The print queue contains:
Print Job ID: 6, Pages: 3

The printer finished printing the job with ID: 5, number of pages: 5 19:09:46
The printer started to print the job with ID: 6, number of pages: 3 (queue size is: 0) 19:09:46
The print queue is empty.

The printer finished printing the job with ID: 6, number of pages: 3 19:09:49
End of the simulation at: 19:09:49

Some Important Rules

Although some of the information is given below, please also read the assignment submission and grading policies from the lecture notes of the first week. In order to get a full credit, your program must be efficient, modular (with the use of functions), well commented and indented. Besides, you also have to use understandable identifier names. Presence of any redundant computation, bad indentation, meaningless identifiers, function names, or missing/irrelevant comments will decrease your grade in case that we detect them.

When we grade your submissions, we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we are going to run your programs in Release mode and **we may test your programs with very large test cases**. Hence, take into consideration the efficiency of your algorithms other than correctness.

How to get help?

You may ask your questions to TAs or to the instructor. Information regarding the office hours of the TAs and the instructor are available at SUCourse.

What and where to submit (PLEASE READ; **UPDATED**; IMPORTANT)

WE WILL NOT USE CODERUNNER FOR THIS ASSIGNMENT!

The internal clock of SUCourse might be a couple of minutes skewed, so make sure you do not leave the submission to the last minute. In the case of failing to submit your work on time:

"No successful submission on SUCourse on time = A grade of 0 directly."

It'd be a good idea to write your name and lastname in the program (as a comment line of course). Do not use any Turkish characters anywhere in your code (not even in comment parts).

If your full name is "Duygu Karaoğlu Altop", and if you want to write it as comment; then you must type it as follows:

// Duygu Karaoglan Altop

Submission guidelines are below. You are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will substantially decrease. Please test your program yourself as any discrepancy in the output may lead to big reduction in your grade.

- You should upload the header and cpp files of the class given to you (PrintQueue) (of course after completing them) as well as your main.cpp ("SUCourseUserName_LastName_OtherNames_THE4.cpp") file.
- Name the main.cpp into:

"SUCourseUserName_LastName_OtherNames_THE4.cpp"

Your SUCourse user name is actually your SUNet username which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SU e-mail address is **duygukaltop@sabanciuniv.edu**, then the file name must be: **"duygukaltop_altop_duygu_THE4.cpp"**.

Wrong file naming will result in points reduction.

- Do not zip any of the documents but upload them as separate files only.
- Please make sure that the files are the latest versions of your assignment program.
- Submit your work **through SUCourse only!** Paper, e-mail or any other methods are not acceptable.

You may visit the office hours if you have any questions regarding submissions.

Please see the previous THEs specifications for the other important relevant rules and the submission guidelines that are not mentioned in this document.