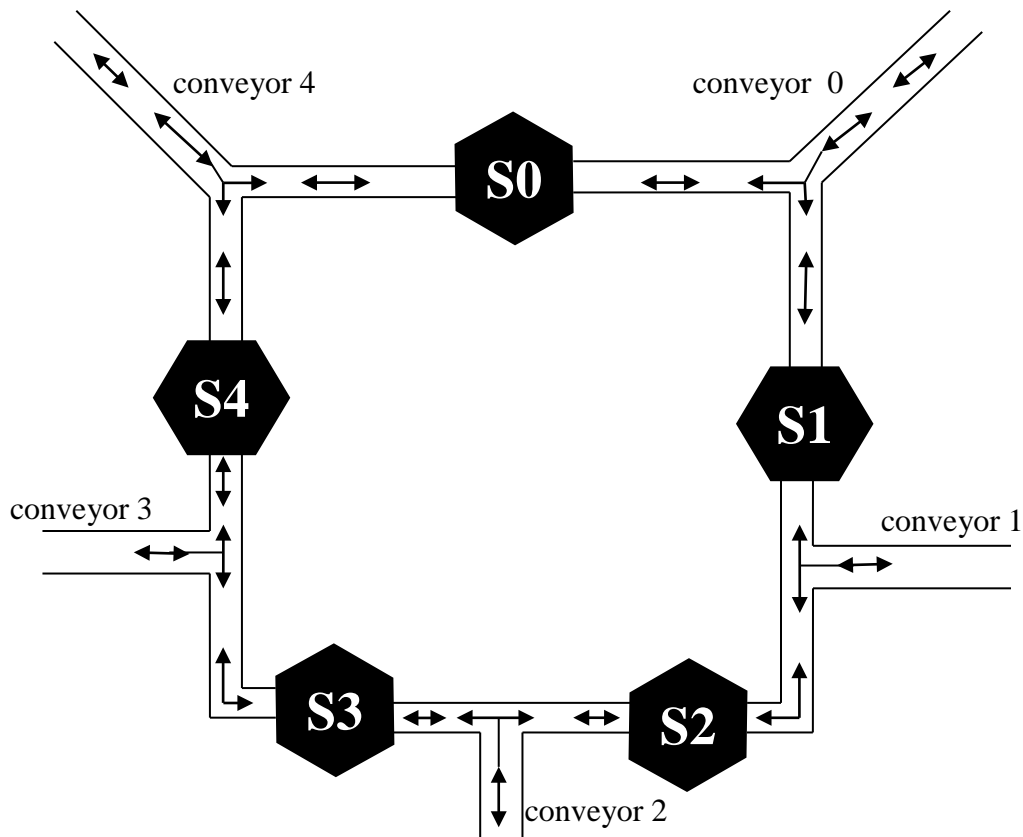**Title:** "Project 2: Multi-threaded Programming in Java"
Points: 100
Due Date: Sunday June 10, 2018 by 11:59pm WebCourses time

Objectives: To practice programming an application with multiple threads of execution and synchronizing their access to necessary shared objects.

Description: In this programming assignment you will simulate the package shipping management system for an automated package shipping operation similar to the one depicted here:

This example package shipping operation has five routing stations (S0 – S4), each of which has an input and output conveyor connecting to conveyor lines (C0 – C4) that go elsewhere in the system. Resources were limited when the system was built so each conveyor going to the rest of the facility must be shared between two routing stations. Since each routing station simultaneously needs an input and output connection to function, access to the shared conveyor lines must be strictly regulated. Flow direction in not important in our simulation.

You have been hired to design a simulator for a new package management system being built with the same design, but possibly fewer/more stations. You are to implement this simulator in Java and have each routing station function in its own thread. A routing station moves packages from one of its connected conveyors to the other. A station's workload is the number of times that a routing

station needs to have exclusive access to the input and output conveyors during the simulation. Once a routing station is granted access to both conveyors it calls its `doWork()` method during which it will flow packages down each of its connected conveyors (of course it must verify that it has access and isn't in conflict with another routing station). After the packages-in and packages-out methods are run, the workload of the routing station is reduced by 1 and the routing station will release both of the conveyors and signal waiting routing stations that the conveyors are available. After executing a flow and releasing its conveyors, a routing station should sleep for some random period of time. A routing station's thread stops running when its workload reaches 0. To prevent deadlock, ensure that each routing station acquires locks on the conveyors it needs in increasing numerical order.

Restrictions:
1. Your source files should begin with comments containing the following information:
   ```
   /*
           Name: <your name goes here>
           Course: CNT 4714 Summer 2018
           Assignment title: Project 2 – Multi-threaded programming in Java
           Date:  June 10, 2018

           Class:  <name of class goes here>
   */
   ```
2. **Do not** use a monitor to control the synchronization in your program (i.e., do not use the Java synchronize statement).

Input Specification:
Your program must initially read from a text file (`config.txt`) to gather configuration information for the simulator. The first line of the text file will be the number of routing stations to use during the simulation. Afterwards, there will be one line for each station. These lines will hold the amount of work each station needs to process (i.e, the number of times it needs to move packages down the conveyor system). Only use integers in your configuration file, decimals will not be needed. You can assume that the maximum number of stations will be 10.

Output Specification:
Your simulator must output the following text to let the user know what the simulator is doing in each of these situations:

1. An input conveyor is set:
   Station X: In-Connection set to conveyor n

2. An output conveyor is set:
   Station X: Out-Connection set to conveyor n

3. A stations workload is set:
   Station X: Workload set. Station X has n package groups to move.

4. A station is granted access to a conveyor:
   Station X: granted access to conveyor n

5. A station is releasing access to a conveyor:

```
           Station X: released access to conveyor n
```

6. A station has completed its workload:
   ```
   * * Station X: Workload successfully completed. * *
   ```

7. A station successfully flows packages down a conveyor:
   ```
   Station X: successfully moves packages on conveyor n.
   ```

8. A station completes a flow:
   ```
   Station X: has n package groups left to move.
   ```
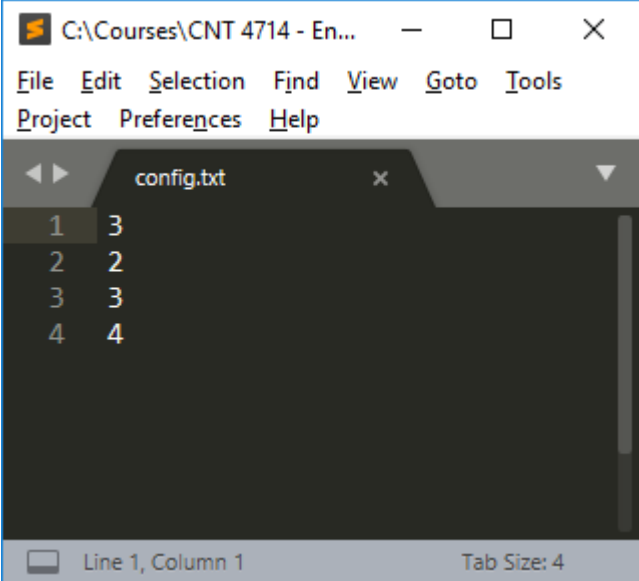
Deliverables:

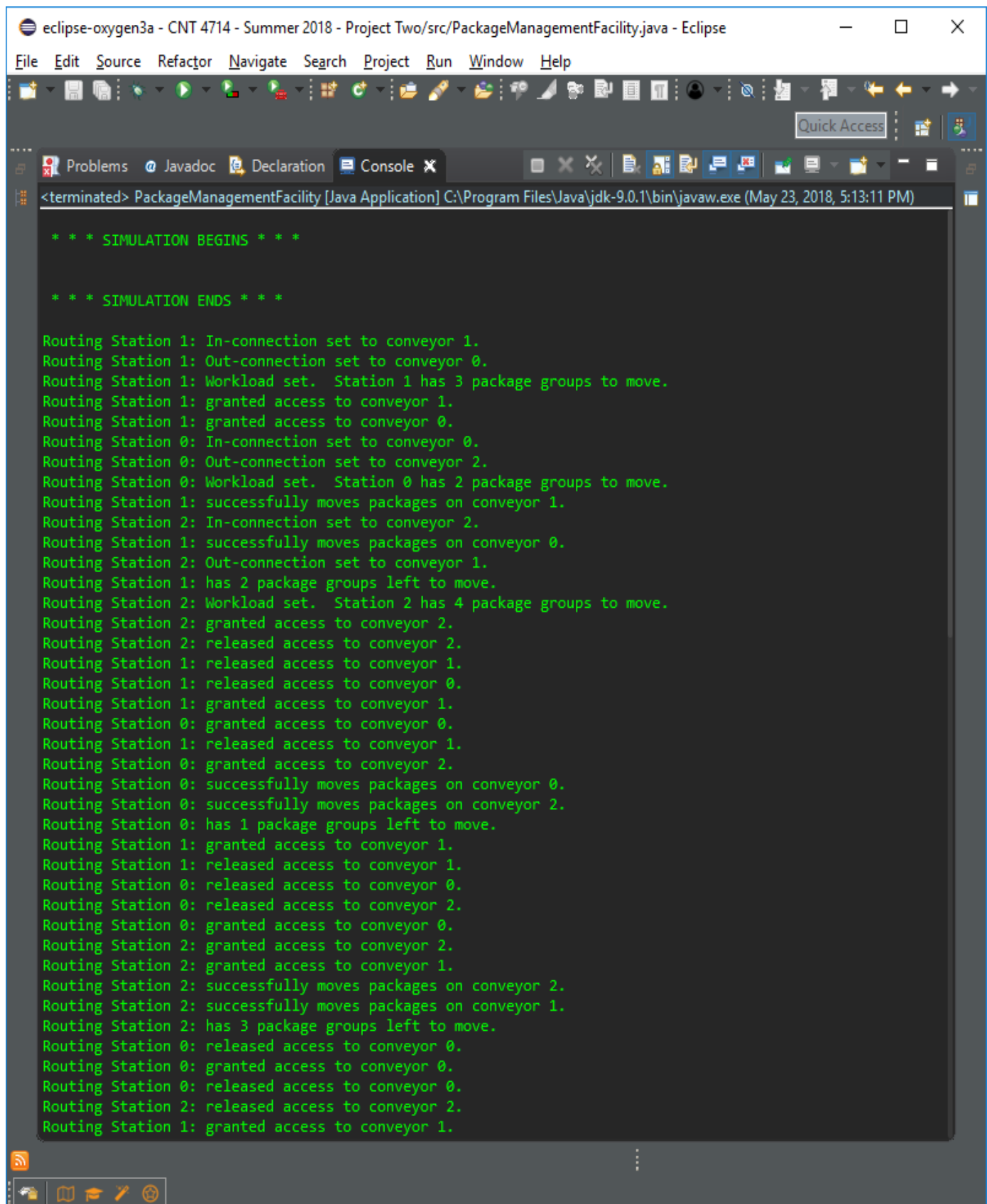Submit the following items via WebCourses no later than 11:59pm June 10, 2018.
      (1) All of your `.java` files.
      (2) A copy of a sample execution of your program, i.e., the output produced by your simulator (this should just be a text file). In Eclipse you can redirect console output to a file, do this and include a copy of the output file produced by your program.

Additional Information:

Actual simulation run in Eclipse (console output redirected in this example) with **config.txt** containing **3 2 3 4**, is shown below.

File   Edit   Source   Refactor   Navigate   Search   Project   Run   Window   Help

Quick Access

Problems   @ Javadoc   Declaration   Console ✕

<terminated> PackageManagementFacility [Java Application] C:\Program Files\Java\jdk-9.0.1\bin\javaw.exe (May 23, 2018, 5:13:11 PM)

```
 * * * SIMULATION BEGINS * * *


 * * * SIMULATION ENDS * * *

Routing Station 1: In-connection set to conveyor 1.
Routing Station 1: Out-connection set to conveyor 0.
Routing Station 1: Workload set.  Station 1 has 3 package groups to move.
Routing Station 1: granted access to conveyor 1.
Routing Station 1: granted access to conveyor 0.
Routing Station 0: In-connection set to conveyor 0.
Routing Station 0: Out-connection set to conveyor 2.
Routing Station 0: Workload set.  Station 0 has 2 package groups to move.
Routing Station 1: successfully moves packages on conveyor 1.
Routing Station 2: In-connection set to conveyor 2.
Routing Station 1: successfully moves packages on conveyor 0.
Routing Station 2: Out-connection set to conveyor 1.
Routing Station 1: has 2 package groups left to move.
Routing Station 2: Workload set.  Station 2 has 4 package groups to move.
Routing Station 2: granted access to conveyor 2.
Routing Station 2: released access to conveyor 2.
Routing Station 1: released access to conveyor 1.
Routing Station 1: released access to conveyor 0.
Routing Station 1: granted access to conveyor 1.
Routing Station 0: granted access to conveyor 0.
Routing Station 1: released access to conveyor 1.
Routing Station 0: granted access to conveyor 2.
Routing Station 0: successfully moves packages on conveyor 0.
Routing Station 0: successfully moves packages on conveyor 2.
Routing Station 0: has 1 package groups left to move.
Routing Station 1: granted access to conveyor 1.
Routing Station 1: released access to conveyor 1.
Routing Station 0: released access to conveyor 0.
Routing Station 0: released access to conveyor 2.
Routing Station 0: granted access to conveyor 0.
Routing Station 2: granted access to conveyor 2.
Routing Station 2: granted access to conveyor 1.
Routing Station 2: successfully moves packages on conveyor 2.
Routing Station 2: successfully moves packages on conveyor 1.
Routing Station 2: has 3 package groups left to move.
Routing Station 0: released access to conveyor 0.
Routing Station 0: granted access to conveyor 0.
Routing Station 0: released access to conveyor 0.
Routing Station 2: released access to conveyor 2.
Routing Station 1: granted access to conveyor 1.
```

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Problems @ Javadoc Declaration Console ✕

<terminated> PackageManagementFacility [Java Application] C:\Program Files\Java\jdk-9.0.1\bin\javaw.exe (May 23, 2018, 5:13:11 PM)

```
Routing Station 1: successfully moves packages on conveyor 1.
Routing Station 1: successfully moves packages on conveyor 0.
Routing Station 1: has 1 package groups left to move.
Routing Station 1: released access to conveyor 1.
Routing Station 1: released access to conveyor 0.
Routing Station 1: granted access to conveyor 1.
Routing Station 1: granted access to conveyor 0.
Routing Station 1: successfully moves packages on conveyor 1.
Routing Station 1: successfully moves packages on conveyor 0.
Routing Station 1: has 0 package groups left to move.

 * * Station 1: workload successfully completed. * *

Routing Station 1: released access to conveyor 1.
Routing Station 1: released access to conveyor 0.
Routing Station 0: granted access to conveyor 0.
Routing Station 0: granted access to conveyor 2.
Routing Station 0: successfully moves packages on conveyor 0.
Routing Station 0: successfully moves packages on conveyor 2.
Routing Station 0: has 0 package groups left to move.

 * * Station 0: workload successfully completed. * *

Routing Station 0: released access to conveyor 0.
Routing Station 0: released access to conveyor 2.
Routing Station 2: granted access to conveyor 2.
Routing Station 2: granted access to conveyor 1.
Routing Station 2: successfully moves packages on conveyor 2.
Routing Station 2: successfully moves packages on conveyor 1.
Routing Station 2: has 2 package groups left to move.
Routing Station 2: released access to conveyor 2.
Routing Station 2: released access to conveyor 1.
Routing Station 2: granted access to conveyor 2.
Routing Station 2: granted access to conveyor 1.
Routing Station 2: successfully moves packages on conveyor 2.
Routing Station 2: successfully moves packages on conveyor 1.
Routing Station 2: has 1 package groups left to move.
Routing Station 2: released access to conveyor 2.
Routing Station 2: released access to conveyor 1.
Routing Station 2: granted access to conveyor 2.
Routing Station 2: granted access to conveyor 1.
Routing Station 2: successfully moves packages on conveyor 2.
Routing Station 2: successfully moves packages on conveyor 1.
Routing Station 2: has 0 package groups left to move.

 * * Station 2: workload successfully completed. * *

Routing Station 2: released access to conveyor 2.
Routing Station 2: released access to conveyor 1.
```