This project is to be developed in groups of two or three. You may choose your partners. On Wednesday 2/13, in class, tell the instructor the names of your partners. In Lab on Friday, 2/15, the instructor will announce group numbers. Group numbers are important because they will be used to access the subversion repository. All code must be committed to the subversion repository prior to having the project graded. To have your project graded, you must set up a 30 minute appointment with your partners and the instructor on Wednesday, Thursday or Friday 3/13-3/15 to demo your software. The instructor is available Wednesday after 2, all day Thursday and All day Friday. There will be no labs on Friday. At that time, the instructor may ask any group member to explain various design choices. Appointments must be made in Lab on Friday 3/8.

1. Create a class called *Interval*. The class should include a long start and end value.
   a. In any constructor that takes a start and/or end value, make sure that start < end.
   b. Overload the stream operators `>>` and `<<` to read and write Intervals. Intervals in a stream should have the format *Interval[start, end]*
   c. Overload the operators `==` and `!=` to mean “start and end are the same” and “start and end are not the same”.
   d. Overload the operators `<` and `<=` to mean “start is less” and “start is less or the same”.
   e. Overload the operators `>` and `>=` to mean “end is greater” and “end is greater or the same”.

   Notice that *Interval[100,1000]* is both less than and greater than *Interval[200,500]*.

2. Create a class called *IntervalSet* which contains one or more intervals
   a. Add a union method that takes an Interval and changes the IntervalSet to cover the new Interval. That is
      ```
      {Interval[0,2], Interval[5,7], Interval[8,9],
      Interval[11,15]).union(Interval[1,8]) yields {Interval[0,9],
      Interval[11,15])
      ```
   b. Overload `+` to combine two IntervalSets to a resulting IntervalSet which covers the same intervals.
   c. Overload `-` to take out the second set of intervals from the first. That is
      ```
      {Interval[0,2], Interval[5,7], Interval[8,9]}-
      {Interval[0,1], Interval[6,10]) yields
      {Interval[1,2], Interval[5,6]}\n      ```

3. Write a program that unit tests your Interval and IntervalSet classes.
4. Write a program that uses your Interval and IntervalSet classes and lets you input people, their departments and weekly availability times (integral hours only). The system then lets you query for times that work for inter-departmental meetings. That is, if you request a meeting with the accounting, printing and mailing departments, the system will give you a (possibly empty) list of times when at least one person from each of those departments can meet. The system should also allow you to change the department for a given person or change that person’s availability (adding times, subtracting times, or replacing the schedule altogether). The system should have some documented menu or set of commands that lets the user intersperse these operations. The program should have some way of saving the data to a file, so that when the program starts, it can, if desired, reload the previous information (a command-line switch would be good for this).

5. For your demo, be prepared to run your unit testing harness. Also, have the following personnel input into a file for the system to load on start-up:

<table>
<thead>
<tr>
<th>Name</th>
<th>Department</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sally Bowren</td>
<td>Advertising</td>
<td>MUWHF 9am-noon</td>
</tr>
<tr>
<td>Joe Schalles</td>
<td>Sales</td>
<td>MWF 9am – 2pm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H 1pm-4pm</td>
</tr>
<tr>
<td>Homer Leroy</td>
<td>Legal</td>
<td>U 8am-5pm</td>
</tr>
<tr>
<td>Lisa Rowan</td>
<td>Legal</td>
<td>MUW 1pm-5pm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H 8am-noon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F 8am-11am</td>
</tr>
<tr>
<td>Timothy Johns</td>
<td>Transportation</td>
<td>MUWHF 7am-3pm</td>
</tr>
<tr>
<td>Lee Benedetti</td>
<td>Executive</td>
<td>MWHF 10am-7pm</td>
</tr>
<tr>
<td>Shelley Albracht</td>
<td>Advertising</td>
<td>UH 9am-5pm</td>
</tr>
<tr>
<td>Scott Burich</td>
<td>Mailing</td>
<td>MUWH 7am-1pm</td>
</tr>
<tr>
<td>Zachary Howard</td>
<td>Transportation</td>
<td>MWF noon-5pm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UH noon-7pm</td>
</tr>
<tr>
<td>Caitlin Rowan</td>
<td>Sales</td>
<td>UH 8am-5pm</td>
</tr>
<tr>
<td>Andy Mann</td>
<td>Sales</td>
<td>MW 11am-5pm</td>
</tr>
</tbody>
</table>

Group 1:
- Braden Hollembaek
- Kevin Beick
- Michael Jason

Group 2:
- Dennis O'Connor
- Andrew Dixon
- Sahil Diwan

Group 3:
- Adam Zucker
- Sam Cagle

Group 4:
- Robert Nels Butler
- John Russel Pridgen
- John Wulf

Group 5:
- John Beck
- Jimmy Cheung
- William Stidhem

Group 6:
- Bryan Adams
- Josh Schmidt
Group 7
  Mike Knowles
  Sean Heuer