1. Study Ben-Ari, Sections 4.1–4.5.

2. Implement Algorithm 3.6: Second attempt in C--.  
   Name your source file alg-3-6.cm. There is no type bool in C--, so you must use an int instead with 0 for false and 1 for true. Define them as follows.

   ```c--
   const int false = 0;
   const int true = 1;
   ```

   With these definitions, the statement `await wantq = false` can be implemented in C-- with the spin lock `while (wantq) ;`. Note that the C standard is to interpret integer 0 as false and any nonzero integer value as true.

   CAUTION: The symbol `p` is a reserved word in C--, so you must change the name of the process from `p` to some other name.

   For your convenience, here is a file named alg-3-6.cm that contains Ben-Ari’s alg-3-2.cm code that you can modify.

   [http://www.cslab.pepperdine.edu/warford/cosc450/alg-3-6.cm](http://www.cslab.pepperdine.edu/warford/cosc450/alg-3-6.cm)

3. Implement Algorithm 3.6: Second attempt in Java.  
   Name your class Alg0306 and your source file Alg0306.java. Insert random delays in such a way that the result is usually incorrect with different results on each run.

   For your convenience, here is a NetBeans project that contains the Java code from Alg0302 that you can modify.

   [http://www.cslab.pepperdine.edu/warford/cosc450/Alg0306.zip](http://www.cslab.pepperdine.edu/warford/cosc450/Alg0306.zip)

   Hand in only the source file Alg0306.java, and not a compressed NetBeans project or a .jar file.