The course introduces a method for planning and developing algorithms and computer programs for parallel processing directly from the specification. In this way, the correctness of the program (i.e., the program satisfies the specifications) is directly derived from the construction steps of the program.

The construction of parallel programs (multiprocess programs) is taught in this course, based on a simple theory in which the need to prove the correctness of each step in the development is known. According to this method, the correctness of the parallel program is fully guaranteed (partial correctness): when the program stops, the values of all the variables will satisfy the specifications.

The course is based entirely on the book "On a Method of Multiprogramming" (Feijen and van Gasteren). In addition to learning the development method, the course (like the book) demonstrates the method with a wide range of parallel algorithms, including parallel programming techniques for handling synchronization issues, deadlocks, and other problems.

The course includes problems and exercises for self-study and implementation of the principles learned.
The following basic topics will be covered:

- A Computational Model
- Program Notation and its Semantics: Hoare triples and weakest-liberal-preconditions
- The Core of the Owicki/Gries Theory: annotating a multiprogram; postconditions
- Private Variables and Orthogonality
- System Invariants
- Mutual Exclusion
- Co-assertions and Strengthening the Annotation
- Synchronization and Total Deadlock
- Individual Progress and the Multibound

As time and progress permit, a modest selection of algorithms will be studied (both in class and home assignments), taken from the following collection (as it appears in the second part of the course textbook, starting at Chapter 13):

- The Safe Sluice
- Peterson’s Two-Component Mutual Exclusion Algorithm
- Handshake Protocols
- Phase Synchronization for Two Machines
- The Parallel Linear Search
- The Initialization Protocol
- The Non-Blocking Write Protocol
- Mutual Inclusion and Synchronous Communication
- A Simple Election Algorithm
- Peterson’s General Mutual Exclusion Algorithm
- Monitored Phase Synchronization
- Distributed Liberal Phase Synchronization
- Distributed Computation of a Spanning Tree
- Shmuel Safra’s Termination Detection Algorithm
- The Alternating Bit Protocol

The home assignments, midterm examination, and final project are all theoretical, in the sense that any amount of programming will be done “on paper”. The assignments and final project are to be prepared and submitted either individually or in pairs.

The final grade will be determined by two homework assignments (10% each), a must-pass midterm examination (20%), and a final project (60%).

The course is exclusively based on the monograph *On a Method of Multiprogramming*, by W.H.J. Feijen and A.J.M. van Gasteren.