

# CMPE 261 - Object Oriented and Concurrent Programming

This course introduces students to object orientation and concurrent processes, using the industrially accepted Java language. Object oriented approach considers the data and the process together. Program and algorithm design are considered under this approach. Object orientation topics include data hiding, inheritance, polymorphism, exception handling and GUI programming. This course also covers programming of concurrent, cooperating sequential processes that include critical sections, mutual exclusion, thread synchronization, semaphores, monitors and deadlock starvation.

## Resources

Online resources for the course can be find at <http://online.bilgi.edu.tr> . Lecture notes and assignments, when appropriate, will also be handled via the online system.

The book “An Introduction to Object-Oriented Programming with Java”, by C. Thomas Wu is recommended as supporting material, but not required. You may check out -although somewhat outdated- lecture notes at <http://e-skolastik.appspot.com/mehmetgencer@yahoo.com/files/OOP-LectureNotes.html> and <http://e-skolastik.appspot.com/mehmetgencer@yahoo.com/files/LSP-LectureNotes.html>, in addition to other material such as lecture slides at <http://e-skolastik.appspot.com/MehmetGencer>

## Assessment

- 20% 1st midterm exam
- 20% 2nd midterm exam
- 30% final exam
- 30% classwork attendance and participation performance

## Outline

1. Background and motivations for object orientation and concurrency
2. Classes, objects, types and -self/mutual- references
3. Inheritance, interfaces, polymorphism, reflection. API design and software lifecycle management

4. Generic behavior in the face of type variance
5. Abstract classes&methods for partial implementations, case studies in class hierarchy design
6. Encapsulation, packages
7. Case studies: AWT
8. Concurrency: using multiple threads of control vs sequential programs
9. Thread monitoring, waiting, and job control
10. Process synchronization in the face of concurrency, deadlocks
11. Case studies
12. Controlling performance by using thread pools
13. Alternative approaches to concurrency management