



University of Nicosia, Cyprus

<b>Course Code</b> COMP-416	<b>Course Title</b> Concurrent Programming	<b>ECTS Credits</b> 6
<b>Department</b> Computer Science	<b>Semester</b> Fall	<b>Prerequisites</b> COMP-212, COMP-354
<b>Type of Course</b> Elective	<b>Field</b> Computer Science	<b>Language of Instruction</b> English
<b>Level of Course</b> 1 <sup>st</sup> Cycle	<b>Year of Study</b> 3 <sup>rd</sup> or 4 <sup>th</sup>	<b>Lecturer(s)</b> Dr Harald Gjermundrød
<b>Mode of Delivery</b> Face-to-face	<b>Work Placement</b> N/A	<b>Co-requisites</b> None

**Objectives of the Course:**

The main objectives of the course are to:

- introduce models, mechanisms, and techniques of concurrent programming
- understand when concurrent programming techniques are appropriate to use and the benefits and risks involved with choosing such an approach
- make aware how to create correct programs using several different concurrent programming mechanisms in different programming languages
- thoroughly discuss concurrent programming paradigms or patterns like threads, safety, liveness, state guarding, atomicity, and locking
- cover in detail object visibility and immutability as well as thread safety and confinement
- provide deep knowledge of handling task executing
- introduce and discuss concurrent programming in functional languages.

**Learning Outcomes:**

After completion of the course students are expected to be able to:

1. analyze problems and apply applicable concurrent solutions when developing software
2. apply concurrent concepts and problem solving techniques
3. apply concurrent programming paradigms and/or patterns for real world problems
4. develop (write/debug/correct) concurrent applications that satisfy safety and liveness properties in various programming languages
5. test concurrent applications for bugs, violation of safety and liveness properties.

**Course Contents:**

1. Introduction to concurrency and its applicability to programming
2. Thread safety
3. Sharing objects in threaded applications
4. Composing objects for a threaded application
5. Concurrent modification of shared data structures
6. Task execution, cancelation, and shutdown

7. Building applications using thread pools
8. Applying threads for GUI elements
9. Liveliness, safety, and testing issues for concurrent applications
10. Synchronization and locking mechanisms and how they are applied
11. Concurrent programming in functional languages.

**Learning Activities and Teaching Methods:**

Lectures, Practical Exercises, Projects and Assignments.

**Assessment Methods:**

Homework, Projects, Mid-term Exam, Final Exam.

**Required Textbooks/Reading:**

Authors	Title	Publisher	Year	ISBN
Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea	Java Concurrency in Practice	Addison- Wesley Professional	2006	9780321349606

**Recommended Textbooks/Reading:**

Authors	Title	Publisher	Year	ISBN
Joe Armstrong	Programming Erlang: Software for a Concurrent World	Pragmatic Bookshelf	2007	978- 1934356005
Jeff Magee, Jeff Kramer	Concurrency: State Models and Java Programs, 2 <sup>nd</sup> Edition	Wiley	2006	978- 0470093559