**********
Course Title: Real-Time: From theory to pratice using the Java Language.
**********

***Objective:***
The proliferation of applications with time constraints has significantly increased the interest for real-time. The range of applications with safety critical constraint is large: avionic control systems, telecommunications, industrial automation and robotics. On the other hand, the compliance with temporal constraints has a direct impact on the performance of applications which are not critical in terms of safety.

Real-Time Java combines the ease of programming in the Java language with the performance required by applications that must conform to real-time constraints. Extensions to the Java language provide features for real-time environments the traditional Java runtime environment is lacking.

***Outline:***
The main objectives of the course are to introduce the basic concepts of real-time scheduling. The course will be divided in three chapters.

***1) Introduction***. We will provide an introduction to the real-time task models and uniprocessor scheduling algorithms. We will describe the periodic and aperiodic job activation models, the fixed and dynamic priority scheduling, the priority assignment rules and optimality.

***2) Feasibility Analysis.*** We will present the cpu load conditions and the calculation of the worst case response time of a task in order to decide if a task is feasible or not (i.e if the task respects its temporal constraints in the worst case conditions).

***3) Resource Sharing and Task Synchronization.*** We will describe the different problems associated with the synchronization of real-time tasks (unbounded priority inversion, interblocking, chain of blocking). We will describe two major algorithms: the Priority Inheritance Protocol (PIP) and the Priority Ceiling Emulation Protocol (PCE).

***Exercises:***

For each of the aforementioned chapters, the course will present the main classes described in the Real-Time Specification for Java (RTSJ). Exercises will be proposed to illustrate these different chapters using the LejosRT environment: a RT-JVM running on a Lego Mindstorms NXT Brick (http://lejosRT.univ-mlv.fr).