Functional Programming

1. **Course number and name:** 020PFSES3 Functional Programming

2. **Credits and contact hours:** 4 ECTS credits, 2x1:15 contact hours

3. **Instructor's or course coordinator's name:** Youssef El Bakouny

4. **Text book:**
   Programming in Scala. Martin Odersky, Lex Spoon, and Bill Venners. 2nd edition. Artima 2010.
   a. **Other supplemental materials:**
      Functional Programming in Scala Specialization. Martin Odersky. Massive Online Open Courses (MOOCs) on Coursera.

5. **Specific course information**
   a. **Catalog description:**
      The goal of this course is to introduce the functional programming paradigm using, mainly, the Scala programming language.
      The course begins with an overview of functional programming followed by a brief explanation of how traditionally imperative languages (such as C ++ and Java) recently incorporated some elements of this paradigm. Then, the course will proceed with the gradual exposition of the evaluation model (used to reason about functional programs) alongside the explanation of the following concepts: recursion and the optimization of recursive functions, the use of functions as values, the partial application of functions, object immutability and its advantages, types and pattern matching, pairs and tuples, lists and functional collections, combinatorial search problem solving using for-expressions, lazy evaluation, functional streams and infinite sequences. These concepts will be illustrated by examples and exercises in Scala.
      Once done, the Java 8 syntax of a subset of these concepts will also be exposed.
      Finally, the course will end with an introduction to program proving using structural induction.

   b. **Prerequisites:** 020POOES1 Object-Oriented Programming

   c. **Required:** Elective for CCE students

6. **Specific goals for the course**
   a. **Specific outcomes of instruction:**
      - Explain the difference between the functional programming paradigm and the imperative programming paradigm.
      - Implement a functional Scala program using the main concepts of the functional paradigm.
      - Analyze a functional program in terms of correctness, maintainability and performance.

- Design and implement a functional program in response to a complex problem.
- Evaluate the quality of a code in terms of maintainability and propose an adequately refactored code.

b. KPI:

KPI	b1	b2	c3	e3	k2	k3
Covered	x	x	x	x	x	x
Assessed	x	x	x	x	x	x

7. Brief list of topics to be covered and approximate lecture hours:

- A comparison of programming paradigms: functional and imperative. An introduction to functional programming and the evaluation model (3 lectures)
- The definition and use of functions. Recursion and the termination of recursive functions. Tail recursive functions. The use of functions as values. Higher Order Functions. Currying and the partial application of functions (4 lectures)
- The definition and use of immutable objects. The review of inheritance and polymorphism in the context of the object-oriented functional programming paradigm offered by the Scala language (3 lectures)
- Types, generics, variance and pattern matching (5 lectures)
- Immutable linked lists and higher order functions on lists. Reduction on lists. Pairs and tuples (4 lectures)
- Immutable collections, the resolution of combinatorial search problems and the use of for-expressions (3 lectures)
- Lazy evaluation, functional streams and infinite sequences (3 lectures)
- Functional programming in Java 8 (1 lecture)
- An overview of the proof of functional programs by structural induction (2 lectures)