

Course Specification Document

Title	Programming Languages
--------------	-----------------------

Credits	5 ECTS
----------------	--------

Aims	<p>This course aims to provide students with comprehensive general knowledge of functional programming languages, their characteristics, and concepts, enabling them to transition from traditional thinking patterns in solving programming problems in imperative languages to a new thinking pattern that allows them to solve problems using any functional language. Additionally, the course aims to distinguish between both patterns in terms of their advantages and disadvantages. Furthermore, it aims to equip students with various knowledge related to concepts and techniques of mobile application design, familiarizing them with different environments for mobile applications, and teaching techniques of linking with data sources and integrating with various advanced services.</p>
-------------	--

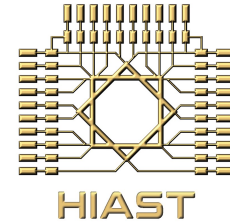
Intended learning outcomes

On successful completion of this course, the student will be able to:

- Know the classifications of programming languages.
- Understand Lambda Calculus computations.
- Understand the philosophy of functional languages through Meta Language (ML) and master it.
- Deal with functional languages and use them in problem-solving.
- Recognize static and dynamic languages.
- Recognize the concepts and techniques of mobile application design.
- Deal with various environments for mobile applications.
- Utilize linking techniques with data sources and the integration techniques with many advanced services.

Syllabus

- **Introduction:** Declarative languages, Imperative languages, Functional languages.
- **Description of λ -calculus computations:** Definition of Lambda Calculus, construction of transformers, expressions and functions, finding the value of an expression or function.
- **Utilization of Lambda Calculus in arithmetic and logical operations.**
- **Construction of recursive functions in Lambda Calculus.**
- **Introduction to Meta Language (ML).**
- **Lists in ML language and recursion.**



- **Enumeration type and User-defined types.**
- **Examples of recursive functions in ML language.**
- **Explanation of Higher Order Functions.**
- **Applying concepts of Encapsulation and Abstraction to functional programming languages through the concepts of Structures and Signatures.**
- **Teaching a dynamic and functional programming language like Lisp.**
- **Comparison between imperative and functional languages.**
- **Practical examples of functional language applications.**
- **Recap of Java basics and Extensible Markup Language (XML).**
- **An introduction to Mobile Applications:** Commercial importance, applications, characteristics.
- **An introduction to the structure of the work environment, the system based on it and its various versions**
- **Activities and event handling.**
- **Core program components:** Intents, services, content providers, manifest.
- **Application interface views:** Buttons, labels, text fields.
- **Application lifecycle.**
- **Linking with data sources:** Concepts of linking with data sources, example of linking with SQLite.
- **Accessing servers and RESTful API applications.**
- **Multi-interface applications and transitioning between them.**
- **Designing advanced services:** Web navigator, GPS, locations.