Algorithms, Data Structures, and Programming Lan-  
guages

Course Description

Programming essentially consists of selecting suitable algorithms and data structures for a specific task and converting them into program code. There are many different programming languages, which are based on different procedures and in which algorithms and data structures are implemented differently. In this module, these concepts, which have so far been dealt with using concrete examples, are systematically presented and applied more broadly in order to give students the necessary tools to develop a systematic approach to programming.

Contents

1. Basic Concepts
   1. Algorithms, Data Structures, and Programming Languages as the Basics of Programming
   2. Detailing and Abstraction
   3. Control Structures
   4. Types of Data
   5. Basic Data Structures (List, Chain, Tree)
2. Data Structures
   1. Advanced Data Structures: Queue, Heap, Stack, Graph
   2. Abstract Data Types, Objects, and Classes
   3. Polymorphism
3. Algorithm Design
   1. Induction, Iteration, and Recursion
   2. Methods of Algorithm Design
   3. Correctness and Verification of Algorithms
   4. Efficiency (complexity) of algorithms
4. Basic Algorithms
   1. Traversing and Linearization of Trees
   2. Search Algorithms
   3. Sorting Algorithms
   4. Search in Strings
   5. Hash Algorithms
   6. Pattern Recognition
5. Measuring Programmes
   1. Type inference and IDE interactive support
   2. Cyclomatic and referential complexity
   3. Digesting code documentation
   4. Compiler optimization
   5. Code coverage
   6. Unit and integration testing
   7. Heap analysis
6. Programming Languages

6.1 Programming Paradigms

1. Execution of Programs
2. Types of Programming Languages
3. Syntax, Semantics, and Pragmatics
4. Variables and Type Systems
5. Overview of Important Programming Languages

7.1 Assembler and Webassembly

1. C and C++
2. Java and C#
3. Haskell and Lisp
4. JavaScript and its relatives
5. Other imperative programming languages