

20CYS312 PRINCIPLES OF PROGRAMMING LANGUAGES L-T-P-C:2-0-3-3

Pre-Requisite(s): 20CYS113 Computer Programming, and 19CSE201 Advanced Programming

Course Objectives

- This course provides a quick overview of different paradigms of programming languages.
- It focuses primarily on the functional programming paradigm using Haskell & Rust.

Course Outcomes

CO1: Understand and implement pure functional programs in Haskell

CO2: Understand and implement programs in Rust

CO3: Formulate abstractions with higher order procedures.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO														
CO1	2	3	3	3	2								2	2
CO2	2	3	3	3	2								2	2
CO3	2	3	2	3								2	3	3

Syllabus

Programming Paradigms - Overview of various Programming Paradigms. Functional Programming with Haskell - GHCi interpreter - functions and types, functional composition, numbers, lists, tuples, type classes, pattern matching, higher order functions: currying, lambdas, maps and filters, folds, IO monad.

Introduction to Rust - Data types, Operators, Decision Making, Loops, Functions, Tuple and Array, Ownership, Borrowing, Slices, Structure, Modules, Collections, Error Handling, File Input and Output, Package Manager, Iterator and Closure, References, Concurrency

Text Book(s)

1. Bird R. *Thinking functionally with Haskell*. Cambridge University Press; 2014.
2. Jim Blandy and Jason Orendorff. *Programming Rust. First Edition*, O'Reilly Media; 2018

Reference(s)

1. *Graham Hutton. Programming in Haskell. Second Edition, Cambridge University Press;2016*
2. *Steve Klabnik, Carol Nichols. The Rust Programming Language. No Starch Press; 2018*

Evaluation Pattern

Assessment	Internal	External
Periodical 1	10	
Periodical 2	10	
Continuous Assessment (Theory) (CAT)	15	
Continuous Assessment (Lab) (CAL)	30	
End Semester		35

*CAT – Can be Quizzes, Assignment, Projects, and Reports.