For students of all Branches [S. Y. /T. Y. /Final year B.Tech. /M. Tech.]

Course Title: Functional Programming in Haskell

Prerequisite: Programming using any high-level language.

Course Objectives:

- **1.** To understand the paradigm of programming known as Functional Programming which is based on the mathematics of lambda calculus.
- **2.** To develop insight about 'lazy' execution.
- **3.** To learn the syntax and semantics of the Haskell programming language.
- 4. To learn 'idioms' of Haskell programming

Course Outcomes: Students will demonstrate the ability to

- **1.** Reason about the correctness of programs.
- **2.** Think in terms of higher-order functions.
- 3. Use data encapsulation and parametric polymorphism.
- **4.** Give importance to the 'type checking' of values/functions and therefore develop programs relatively faster.

Course Contents:

Module	Details	Hours
1	Types and Values	4
2	Functions, Type Inference, Recursion	4
3	Higher-order Functions, Polymorphic Types, Lambda Functions	4
4	Algebraic Data Types, Type Classes	4
5	Recursive Data Types	4
6	I/O	4
7	Advanced Concepts: Functors, Monads	4

There will be programming tutorial for every topic mentioned in the 'Details' column.

Text Books:

1. Brian O'Sullivan, John Goerzen and Don Stewart, 'Real World Haskell', O'reilly.

2. Miran Lipovača, 'Learn You a Haskell for Great Good!', No Starch Press.

Evaluation pattern will be described by the concerned course coordinator during the course.