UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 8 Examination in Engineering: January 2026

Module Number: EC8206 Module Name:Functional Programming Part-B

[Two Hours]

[Answer **all questions**, each question carries 10 marks]

Instructions for candidates

- You must attempt Part A first.
- Part A will be collected after one hour.
- **Once Part A is submitted, you may proceed to Part B, which has a time allocation of 2 hours.**

Q1. a) Explain the fundamental principles of functional programming, focusing on **pure functions** and **immutability**. Illustrate how these principles influence program behavior and reliability.

[2.0 Marks]

b) Compare functional programming with object-oriented programming. Your answer should discuss differences in:

- state management
- control flow
- data transformation

[2.0 Marks]

c) Discuss why functional programming matters in engineering practice. Explain its role in:

- program correctness
- concurrency and parallelism
- data processing pipelines

[2.0 Marks]

d) Explain the concepts of **concurrency**, **parallelism**, and **profiling** in the context of functional programming. Briefly discuss how FP language features support safer concurrent and parallel execution. [2.0 Marks]

e) What are the types of the following functions

- second xs = head (tail xs)
- twice f x = f (f x)

[2.0 Marks]

Q2. a) Give four possible definitions for the logical and operator (&&) using pattern matching.

[2.0 Marks]

b) A triple (x,y,z) of positive integers is called pythagorean if $x^2 + y^2 = z^2$. Using a list comprehension, define a function

```
pyths :: Int → [(Int, Int, Int)]
```

that maps an integer n to all such triples with components in [1..n]. For example:

```
pyths 5 = [(3,4,5), (4,3,5)]
```

[3.0 Marks]

c) Explain recursion in the following function:

```
reverse :: [a] -> [a]
reverse [] = []
reverse (x:xs) = reverse xs ++ [x]
```

For the input: `reverse [1, 2, 3, 4]`

[3.0 Marks]

d) Explain what higher-order functions are in functional programming and illustrate with an example.

[2.0 Marks]

Q3. a) Redefine **map f** and **filter p** using **foldr** in Haskell.

[2.5 Marks]

b) Using interactive programming write a simple function in Haskell that prompts for a string to be entered and displays its length

[2.5 Marks]

c) Using recursion and the function add, define a function that multiplies two natural numbers.

[2.5 Marks]

d) Explain the term lazy evaluation in Haskell with its key features

[2.5 Marks]