# 1.1 – Programming Paradigms

# 1.2 – Course Details

# 2.1 – What is a pure function?

# 2.2 – What is functional programming?

# 3.1 – Getting Started with Haskell

# 3.2 – Writing Our Own Functions

# 4.1 – If

# 4.2 – Let

# 5.1 – Tuples

# 5.2 – List

# 6.1 – List Ranges

# 6.2 – List Comprehensions

# 7.1 – Recursion

# 7.2 – More Complex Recursion and Guards

# 8.1 – List Recursion

# 8.2 – List Recursion Examples

# 9.1 – Where Syntax

# 9.2 – Recursion with multiple lists

# 9.3 – Mutual and Multiple Recursion

# 10.1 – The Caesar Cipher

# 10.2 – Cracking the Caesar Cipher

# 11 – *Assignment 1*

# 12.1 – Types

# 12.2 – Function Types

# 13.1 – Polymorphic Types

# 13.2 – Type Classes

# 14.1 – More Type Classes

# 14.2 – Higher Order Functions

# 14.3 – Anonymous Functions

# 15.1 – Map

# 15.2 – Filter

# 16.1 – Fold

# 16.2 – More on Fold

# 17.1 – Scan

# 17.2 – More Higher Order Functions

# 18.1 – Higher Order Programming Example

# 18.2 – Voting Examples

# 19.1 – Custom Types

# 19.2– More Complex Custom Types

# 20.1 – *Assignment 1 Feedback*

# 20.2 – *Assignment 2*

# 21.1 – Parameterized Custom Types

# 21.2 – Maybe and Either

# 22.1 – Recursive Types

# 22.2 – Trees

# 23.1 – IO

# 23.2 – Writing IO Code

# 24.1 – Writing Programs

# 24.2 – Useful IO Functions

# 25 – IO Example

# Class Test Information

# 26.1 – Evaluation Strategies

# 26.2 – Lazy Lists

# 27 – Tail Recursion

# 28.1 – *Assignment 2 Feedback*

# 28.2 – *Assignment 3*