Course 10975-A: Introduction to Programming

1. **Introduction to Core-Programming Concepts**
    a. Computer Data Storage and Processing
    b. Application Types
    c. Application Life-Cycles
    d. Code Compilation

    Lab: Thinking Like a Computer

    a. Describe Computer Data Storage and Processing Concepts
    b. Describe Application Types
    c. Describe the lifecycle of Application
    d. Describe Code Compilation

2. **Core Programming Language Concepts**
    a. Syntax
    b. Data Types
    c. Variables and Constants

    Lab: Working with Data Types

    a. Define Syntax
    b. Explain the different types of Core Data used in Programs
    c. Declare a Use Variables and Constants in a Computer Program

3. **Program Flow**
    a. Introduction to Structured Programming Concepts
    b. Introduction to Branching
    c. Using Functions
    d. Using decision Structures
    e. Introduction Repetition

    Lab: Creating Functions, Decisions and Looping

    a. Describe Structured Programming
    b. Create and Use function in your code
    c. Create and use DE scion Structures
    d. Create and use looping structures

4. **Algorithm and Data Structures**
    a. Understand How to write Pseudo Code
    b. Algorithm Examples
    c. Introduction to Data Structures

Lab: Working with Algorithms and Data Structures

a. Transfer problem statements into pseudo code
b. Create Algorithms
c. Translate Pseudo code in to programming code
d. Create simple algorithm in code
e. Create data structures to store data

5. Error Handling and Debugging

a. Introduction to Program Errors
b. Introduction to Structed Error Handling
c. Introduction to Debugging in Visual Code

Lab: Implementing Debugging And Error Handling

a. Implement Structured Exception Handling
b. Debug Application

6. Introduction to Object Oriented Programming

a. Introduction To Complex Structures
b. Introduction to Structs
c. Introduction Classes
d. Introducing Encapsulation

Lab: Implementing Complex Data Structures

a. Create and Use Structure Types
b. Create and use basic class files
c. Choose when to use a struct vs a Class

7. More Object-Oriented Programming

a. Introduction to Inheritance
b. Introduction to Polymorphism,
c. Introduction to .Net Framework and the Base Class Library

Lab: Implementing Polymorphism

a. Use Inheritance in OOP
b. Implement polymorphism in your classes
c. Describe how the base class library is constructed
d. Find class information by using the Object Browser

8. **Introduction to Application Security**
    a. Authentication and Authorization
    b. Code Permissions on Computers
    c. Introducing Code-Signing

9. **Core I/O Programming**
    a. Using Console, I/O
    b. Using File, I/O

Lab: Core I/O Programming

a. Read input from a Console
b. Output Data to the console
c. Read and Write Text Files

10. **Application Performance and Memory Management**
    a. Value Types vs Reference Types
    b. Converting Types
    c. The Garbage Collector

Lab: Using Value Types and references Types

a. Implement Value and Reference types Correctly in an application
b. Convert between value types and Reference Types
c. Use the garbage Collector