Syllabus

1. Foundations of IT Fundamentals (computer science)

In this module, you will understand core functionalities of Windows, macOS, and Linux operating systems. Customize web browser settings to enhance security and browsing performance. Software tools for productivity. *Skill you will gain:* Application Development, Operating Systems, Information Technology, Programming Principles.

OS: Introduction to the Specialization, Computer Operating Systems, Microsoft Windows, Apple macOS, GNU Linux

Software Applications: Software Usage Requirements, Web Browser Settings, Browser Security, Browser Privacy

Computer Network: Introduction to Networking Protocols & Services, Introduction to Cloud Computing, Information Security

Graded Practice Quiz: For this quiz, you will answer questions based on your knowledge (30) with time management(50min).

Guided Lab: you try lab with introduction of task and editor.

2. Foundation of Coding Backend

In this module, you will understand the core principles of back-end development and the role of back-end engineers. By the end, you will create a simple project plan and a version control repository on GitHub, applying your knowledge to practical scenarios. *Skill you will gain:* use of Git and GitHub for version control, Microsoft Copilot for Debugging and Logic Error, Back-End Development Principles, Debugging Techniques.

Introduction to backend development : Introduction to Back-End Development Principles, Roles and Responsibilities of a Back-End Engineer, Skills , Technologies and Workflow in Back-End Engineering, Project Planning Fundamentals for Back-End Development, Resource Management and Documentation in Back-End Projects, Tool Integration for Back-End Project Planning

Project development : Integrated Coding Practice, Introduction to Debugging, Debugging Techniques, Introduction to Microsoft Copilot, Microsoft Copilot for Debugging and Logic Error, Basics of Git for Back-End Development. How Does the Internet works?, DNS, Hosting, How does browser works?, HTTP vs HTTPS.

Graded Practice Quiz: For this quiz, you will answer questions based on your knowledge (30) with time management(50min).

Guided Lab: you try lab with introduction of task and editor.

3. Algorithms

In this module, you will explore essential skills in data structures, algorithms, and performance optimization for back-end development. *Skill you will gain:* fundamental data structures, sorting and searching techniques, and advanced algorithmic strategies, hands-on implementation, performance analysis, and the use of Microsoft Copilot.

Introduction to Logical Thinking and Problem-Solving: Types of Logic Used in Programming , Deductive Reasoning in Programming, Steps in Deductive Reasoning, Problem Decomposition, Techniques for Problem Decomposition, Top-Down Problem-Solving Approach, Bottom-Up Problem-Solving Approach, Comparing Top-Down and Bottom-Up Approaches, Introduction to Pseudocode

Flowcharts + basic DataTypes: Introduction to Algorithms, Flowcharts, and Data Types, Basic/ Practical Algorithm Structures, Basics / Practical / Analyzing of Flowcharting, Fundamental Data Types, Data Type Conversion, Introduction to Variables, Declaring Different Data Types, Variable Declaration, Initialization and Assignment Keywords, Algorithm Design

Fundamentals of Control structures and Loops, Functions, Methods, Scopes: Introduction to Control Structures and Loops. Introduction to If-Else Statements; Advanced If-Else Statements. Switch Statements; Complex Switch Cases. Control Structures, Decision-Making Scenarios. For, While, Do-While loops. Combining Loops and Control Structures. Loop Optimization. Integrating If-Else and Switch Statements With Loops. Integrated Use of Control Structures and Loops. *Introduction to Functions in Programming. Writing and Using Functions in Programs. Introduction to Methods. Syntax of Methods. Use Cases for Method. Basics of Calling Methods. Method Return Values. Parameters. Passing Data into Methods. Integrating Functions and Methods. Solving Problems With Functions and Methods. Developing Programs with Functions and Methods

Introduction data structure: Characteristics of Arrays and Linked list. Use case for Stack and Queue. Choosing the Right Data Structure for the Job, Implementing Arrays and Linked Lists, Introduction to Big O Notation and applying to Data Structures

Sorting and Searching algorithms: Introduction to Sorting and Searching Algorithms, Introduction/Implementation to Bubble Sorting Algorithms, Mechanics of Quicksort and Merge Sort and their Implementation. Implementing and Testing Sorting Algorithms, Linear Search in Data Structures, Applying Linear Search in Data Structures, Binary Search in Sorted Data Structures, Applying Binary Search in Sorted Data Structures, Implementing Binary Search , Applying Binary Search in Back-End Systems, Code Optimization Techniques Using Binary Search, Identifying Best-Use Cases for Sorting Algorithms in Back-End Systems

Trees and Graphs: Introduction to Trees and Graphs, Practical Use Cases for Tree Structures, Introduction/ Implementation to Tree Traversal Techniques, Directed vs. Undirected

Graphs, Implementing Graph Traversal Algorithms. Introduction to Load Balancing Algorithms and Strategies.

Advanced Problem solving: Introduction to Advanced Algorithms and Problem-Solving. Overview of Dynamic Programming and Greedy Algorithms. İntroduction to Code-Based Dynamic Programming Task. Implement Dynamic Programming in a Scheduling Application. Introduction to Hashing Concepts. Demonstrating Hash Table Implementation. Applying Copilot for Comprehensive Performance Optimization

Graded Practice Quiz: For this quiz, you will answer questions based on your knowledge (30) with time management(50min).
Guided Lab: you try lab with introduction of task and editor.

4. Introduction to Programming C#

In this module, you will learn both the fundamentals and advanced aspects of C# programming and gain hands-on experience setting up projects and using object-oriented principles to create scalable applications. *Skill you will gain:* CLR, Garbage Collector, Memory types, Lambda Expression, LINQ, Collections, OOP fundamentals, Thread/Tasks, Exception handling, Delegate/Events, Reflection, Stream class, introducing new version of C sharp 7-13.

Fundamentals: C# vs C programming, C# basics (statement, syntax, comments, namespace, top-level statement, global using directive, directive vs statement, declaration, Object class), CLR,CTS,CLS ; Managed vs Unmanaged code. Assembly, Exe, DLL , GAC, Nuget package. Application domain vs Process vs Thread, Inter process communication (pipes/shared memory,sockets) vs Intra process communication (lock/mutex/events/condition), Garbage Collector (close, dispose, finalize/destructor, generations, using statement).

Datatypes: Var vs Object vs Dynamic; Implict vs Primitive types; Float vs Double vs Decimal

Operators: Assignment operator, then arithmetic operator including increment and decrement, relational operators, logical / bitwise / equals() / == operators, unary operators.

Memory types: Value and Referance types, Stack vs Heap, String vs StringBuilder, String Format vs Interpolating, Substring, Boxing vs Unboxing, Implict vs Explict types, Casting (upcasting vs downcasting), Convert vs Parse vs TryParse, Type Checking (is/as), Nullable value type/Null forgiving/Null coalescing, Ref/Out/Ref Local/Ref Return/In

Control statements: Selection statement, Iteration statement, Ternary, Const and Readonly.

Methods: what is Method, Extension, Expression, Tuple, Nested methods, Overload/Override methods,

OOP: what is OOP, Class/Record/Struct/Enum, Field, Property (get/set/init), Constructor(private/static/chaining), private vs static constructor, Deconstructor, Base/This

keys, Static class, Anonymous class, Sealed class, Access modifiers, Polymorphisim, Encapsulation, Virtual and Hiding methods, Abstract class and Abstract methods. Inheritance vs Interface (include changes 8/11), Abstract vs Interface

Arrays: What are Arrays, Single/MultiDimensional arrays, Jagged Arrays, Object array, Dynamic array. Indexer and Partial class.

Generics and Collection: what is Generics (generic class/method/interface), Collection(generic vs non-generic collection), Non-generic types (arraylist,Hashtable,sortedlist,stack/queue) , Array vs ArrayList, Generic types (List<T>(), Dictionary<Tkey,Tval>(), SortedDictionary<Tkey,Tval>(), Stack<T>/Queue<T>() , HashSet<T>(), LinkedList<T>()), DicionaryEntry vs KeyValuePair, List<T>() vs LinkedList<T>(),

Why we use Collections, Collection Initializers, IEnumerable(Enumerator) vs ICollection vs IQueryable, Advantages vs Disadvantages of generics,

LINQ and Lambda : What is LINQ, Query vs Method syntax, LINQ operators (select/selectmany/orderBy/First/Single/Last/All/Any/ThenBy/Skip/SkipWhile/Take/TakeWhile) , TakeWhile vs Where clause, Where vs Let clause, AsEnumerable vs AsQueryable method in LINQ, How LINQ with DB can be used?, The role of DataContext class in LINQ, Why SELECT clause comes after FROM clause?, How LINQ is useful more than StoredProcedure?, 3 main concepts of LINQ, ühat is LINQ provider? , Query Expression, Types of LINQ (LINQ to Object/ LINQ to XML/ LINQ to SQL/ LINQ to DataSet/ LINQ to Entites), DataContext vs DbContext. Architecture of LINQ, PLINQ, Advantages and Disadvantages of LINQ,

What is Lambda expression?, Where we use Lambda expression?, What are important point to remember while using Lambda?, Expression vs Statement Lambda, Type interface in Lambda, Rules of Lambda, Does Lambda expression do have a type?

Error and Exception: *Information about content:* What is an Error?, Type of Error (compile time, run time, logical error), What is an Exception?, Classes in Exception (DividingByZero, SQLException, FormatExcept etc), THROW vs THROW EX usage, Exception Handling (try/catch/finally), System.Exception vs System.Application.Exception, Inner Exception, Aggregate Exception, Error vs Exception.

Async programming: What is Thread, Types of Threads, Thread vs Process (repeated above), Thread Pool, Multithreading, Async vs Sync mechanisim, Thread.Sleep() , Thread.Join(), Thread.Abort(), Thread.IsAliv() methods usage, Lock/Monitor/Mutex, Inter and Intra process communication between threads, Deadlock, Race Condition. What is Task and method: Task.Run() vs Task.FromResult(), Task.WhenAll(), Task.Delay(), Task.WaitAll(), Task.ContinueWith(), Async vs Multithreading, I/O bound vs CPU bound, Context Switching, Interlocked, Async and Await keys, Paralel programming and methods: Paralel.For() vs Paralel.ForEach(), Paralel.Invoke() methods. Async vs Paralel, Multithreading vs Paralel.

Delegate and Event: What is Delegate and why is used?, Multicast delegate, Delegate invocation, Callback functions, When we use delegate instead of Interface?, How will we handle Exception in MulticastDelegate?, BeginInvoke vs EndInvoke, Overloading vs Delegate, BuiltIn delegates, Generic vs Custom Delegates, How Delegate Interface and mehtod group conversion used to simplfy delegate instantiations?, Anonymous method vs Lambda expression.

What is an Event, Classes of Event (Subscriber/Publisher), what is EventHandling? EventHandler method, Async Event Handling,

Reflection: what is Reflection, Typeof() vs GetType(), When reflection is useful, What are the risk of using Reflection?, Binding process (binding a type, binding flags), MehtodInfo /PropertyInfo usages, Late Binding vs Early Binding

Stream: What is Stream class, Properties (CanRead,CanWrite,CanSeek,Length,Position) and Mehtods (Read(),Write(),Seek(),Close(),Flush()), Seek vs Position, DerivedClassess (FileStream, StreamReader/StreamWriter, BinaryWriter,BufferedStream), AsyncOperations (ReadAsync, WriteAsync), What is Buffer, Chunk Reading/Writing, Handle Exception in Stream, Manage Resources, Security/Access control, Handle Reading/Writing. File vs FileInfo classes, Directory vs DirectoryInfo classes.