Module 1

What to learn

- Introduction
- Understanding Visual Studio IDE
- Creating Hello World Application
- Debugging
- Datatype
    - Value Type
    - Reference Type
- Variables
- Arrays
- If statement/Switch
- Operators
- Looping Constructs
- Constants (enum)

Practice Exercise

Practice 1

Do the hands on the things provided in videos and provided in the following url https://docs.microsoft.com/en-us/visualstudio/get-started/csharp/tutorial-console?view=vs-2019

Practice 2

Introduction to C# and Visual Studio IDE

Explore Visual Studio:

Open Visual Studio, create a new C# Console Application, and navigate the Solution Explorer, Properties, and Output Window.

Hello World Application:

Create a Console Application that prints "Hello, World!" to the console.

Modify Hello World:

Update the program to ask the user for their name and greet them with "Hello, [Name]!".

Using Comments:

Add single-line and multi-line comments to the Hello World program explaining each part of the code.

Building and Running:

Build and run the Hello World application, observing the build output and errors if any.

Practice 3

Debugging

Adding Breakpoints:
Add a breakpoint to the Hello World application. Debug it and observe the flow of execution.

Step Over/Into:
Write a program with a simple function and use "Step Into" and "Step Over" in the debugger.

Variable Watch:
Declare and initialize a few variables. Use the debugger to watch their values during execution.

Error Simulation:
Introduce a deliberate error (like a division by zero). Debug the program to understand the error.

Debugging Logic Errors:
Write a program to calculate the sum of two numbers. Intentionally make a logic error and debug it to fix the issue.

Practice 4

Data Types

Value Types:
Write a program declaring int, float, and char variables. Assign and print their values.

Reference Types:
Create a program using a string and an array. Assign values and print them.

Type Conversion:
Perform explicit and implicit type conversions between int and double.

Nullable Types:
Create a nullable int variable, assign values, and handle cases where it is null.

Dynamic Type:
Write a program to declare a dynamic variable, assign different types of values, and print them.

Practice 5

Variables

Declaring Variables:
Declare and initialize variables of different types, and print them using string interpolation.

Arithmetic with Variables:
Write a program to perform arithmetic operations using two variables.

Constant Variables:
Declare a const variable for the value of Pi and use it to calculate the circumference of a circle.

Scope of Variables:
Write a program demonstrating local and global variable scopes.

Swapping Variables:
Write a program to swap the values of two variables without using a third variable.

Practice 6

Arrays

Single-Dimensional Array:
Write a program to store 5 student marks in an array and display them.

Multi-Dimensional Array:
Create a 2D array of integers and initialize it with values. Print its elements using nested loops.

Jagged Array:
Create a jagged array to store the names of students in different classes and print them.

Array Length:
Write a program to find the length of an array entered by the user.

Array Sorting:
Create an array of integers, sort it in ascending order, and display the sorted values.

Assignment Exercise

Assignment 1 (65cdfdab7ba36e06448762c8)
Print sum of all the even numbers
Assignment 2 (65cdfdab7ba36e06448762c9)
Store your name in one string and find out how many vowel characters are there in your name.
Assignment 3 (65cdfdab7ba36e06448762ca)
Create a weekday enum and accept week day number and display week day.
Assignment 4 (65cdfdab7ba36e06448762cb)
Accept 10 student Name,Address,Hindi,English,Maths Marks ,do the total and compute Grade. Note do it with Array and display the result in grid format
Assignment 5 (65cdfdab7ba36e06448762cc)
Accept Age from user, if age is more than 18 eligible for vote otherwise it should be displayed as not eligible. Do it with ternary operator
Assignment 6

Assignment: Employee Management System

Create a **Console Application** in C# that serves as a basic **Employee Management System**. The program should integrate the concepts learned in **Introduction to C#, Debugging, Data Types, Variables, Arrays, Conditional Statements, Loops, and Constants/Enums**.

Objective:

Build a program that:

> Allows users to manage employee records (add, view, search, and calculate salaries). Includes logic to determine bonuses based on performance ratings.

Requirements:

1. Introduction to Visual Studio & Debugging:

> **Create the project**: Start a new Console Application in Visual Studio named EmployeeManagementSystem.
> **Use Debugging tools**: Add breakpoints and debug the application as needed.

2. Employee Class (Data Types, Variables, and Reference Types):

Create a class Employee with the following properties:

int ID (Value Type)
string Name (Reference Type)
string Department
double Salary
int PerformanceRating (1-5 scale)

3. Menu (If Statements/Switch):

Create a menu-driven system with the following options:

Add an Employee.
Display All Employees.
Search Employee by ID.
Calculate Bonus for Employees.
Exit.

4. Business Logic (Operators and Arrays):

Add Employee: Store employee details in an array or a list (use an array of size 10 for simplicity).
Display All Employees: Loop through the array and display details of each employee.
Search Employee by ID: Use a loop to find the employee with the matching ID and display their details.
Calculate Bonus: Use this formula: Bonus=Salary×(Performance Rating10)\text{Bonus} = \text{Salary} \times \left(\frac{\text{Performance Rating}}{10}\right)Bonus=Salary×(10Performance Rating) Display each employee's bonus.

5. Constants/Enums:

Use an **enum** for department names (e.g., HR, IT, Finance, Sales).
Use a **constant** for the base salary multiplier (e.g., const double BaseMultiplier = 0.1;).

6. Validation (Conditional Statements):

Validate user input (e.g., Performance Rating should be between 1 and 5).
Ensure the array doesn't exceed its size (handle cases when trying to add more than 10 employees).

Welcome to Employee Management System
---
1. Add Employee
2. Display All Employees
3. Search Employee by ID
4. Calculate Bonus for Employees
5. Exit

Enter your choice: 1

Enter Employee ID: 101
Enter Name: John Doe
Enter Department (HR/IT/Finance/Sales): IT
Enter Salary: 50000
Enter Performance Rating (1-5): 4

Employee Added Successfully!

Enter your choice: 4
ID: 101, Name: John Doe, Bonus: 2000

Deliverables:

Submit the C# project folder containing the complete solution.
Include the following:

Employee.cs: Class definition for Employee.
Program.cs: Main program logic

Online Reference

https://docs.microsoft.com/en-us/visualstudio/get-started/csharp/tutorial-console?view=vs-201

https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/

https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/

.NET Core Web API

WEB API (old)

Authentication And Authorization (WEBAPI)(old)

FullStackDevelopment_WIth_Dotnet_AND_Angular