****Course Title: Introduction to Modern C++ Programming****

****Session 1: Introduction to C++ and Development Environment Setup****

* Brief history of C++
* Installing a C++ development environment (e.g., Visual Studio, Code::Blocks, or an online IDE)
* Writing and running a simple "Hello World" program
* Basic concepts: source code, compiler, and linker
* Assignment: Write a program to display a personalized greeting.

****Session 2: Variables and Data Types****

* Variables, data types, and memory allocation
* Fundamental data types (int, float, double, char)
* User-defined data types (struct)
* Variable declaration, initialization, and assignment
* Input and output using **cin** and **cout**
* Assignment: Create a program to calculate the area of a rectangle.

****Session 3: Control Structures****

* If statements and conditional branching
* Switch statements
* Loops (for, while, do-while)
* Logical operators (&&, ||, !)
* Assignment: Implement a simple calculator with options for addition, subtraction, multiplication, and division.

****Session 4: Functions and Modularity****

* Functions and their syntax
* Function parameters and return values
* Function overloading
* Scope and lifetime of variables
* Assignment: Write a program to find the maximum and minimum of three numbers using functions.

****Session 5: Arrays and Strings****

* Arrays and their declaration
* Array indexing and iteration
* Introduction to C-style strings
* Basic string manipulation
* Assignment: Create a program to reverse a string using arrays.

****Session 6: Pointers and References****

* Understanding memory addresses
* Pointers, pointer declaration, and initialization
* References and their benefits
* Pointers and arrays
* Assignment: Implement a function to swap two numbers using pointers.

****Session 7: Object-Oriented Programming (OOP) Basics****

* Introduction to OOP concepts
* Classes and objects
* Member variables and member functions
* Constructors and destructors
* Assignment: Create a class for a simple calculator and demonstrate its usage.

****Session 8: Inheritance and Polymorphism****

* Inheritance and its types (single, multiple)
* Base and derived classes
* Polymorphism and virtual functions
* Assignment: Extend the calculator example from the previous session with multiple calculator types (e.g., scientific, simple).

****Session 9: Standard Template Library (STL)****

* Introduction to the C++ STL
* Containers (vector, list, map)
* Iterators and algorithms
* Assignment: Create a program that uses a vector to store and manipulate a list of numbers.

****Session 10: Exception Handling****

* Understanding exceptions
* try, catch, throw, and noexcept
* Exception hierarchies
* Assignment: Develop a program that handles input validation using exceptions.

****Session 11: File Handling****

* Reading and writing text files
* File streams (ifstream, ofstream)
* Error handling for file operations
* Assignment: Create a program to read data from a text file and display it.

****Session 12: C++ Best Practices and Modern Features****

* C++11, C++14, and C++17 features
* Smart pointers
* Lambda expressions
* Assignment: Revise any previous assignment to incorporate modern C++ features.

****Session 13: Final Project****

* Students work on a project of their choice, applying the knowledge gained throughout the course. This could be a small game, a database application, or any other application.

****Session 14: Course Review and Q&A****

* Final session for reviewing key concepts, answering questions, and providing guidance for further learning.