

Title: C++ Language with Data Structures Course Syllabus

Introduction: Welcome to the C++ Language with Data Structures course. This syllabus is designed to provide students with a comprehensive understanding of the C++ programming language along with the implementation of fundamental data structures. Throughout this course, students will learn the syntax, semantics, and best practices of C++ programming, as well as gain proficiency in implementing and utilizing various data structures for efficient problem-solving.

Course Objectives:

1. To introduce students to the fundamentals of the C++ programming language.
2. To familiarize students with the concepts and implementation of essential data structures.
3. To enable students to analyze problems and design algorithms using data structures.
4. To provide practical experience in coding and debugging C++ programs with data structures.
5. To enhance problem-solving skills through hands-on programming exercises and projects.

Course Outline:

1. Introduction to C++ Programming
  - Overview of C++ language features and benefits
  - History and significance of C++
  - Basic structure of a C++ program
  - Variables, data types, and operators
  - Input and output operations using streams
2. Object-Oriented Programming (OOP) Concepts
  - Principles of object-oriented programming
  - Classes and objects
  - Encapsulation, inheritance, and polymorphism
  - Constructors and destructors
  - Friend functions and operator overloading
3. Control Structures and Functions
  - Decision making with if-else and switch statements
  - Looping constructs: while, do-while, and for loops
  - Function declaration and definition
  - Passing arguments to functions by value and reference
  - Function overloading and default arguments
4. Pointers and Dynamic Memory Allocation

- Understanding pointers and memory management in C++
- Pointer arithmetic and pointer expressions
- Dynamic memory allocation using new and delete operators
- Smart pointers and memory management best practices

#### 5. Arrays, Strings, and Standard Template Library (STL)

- Arrays and array manipulation
- Introduction to string class and string manipulation
- Overview of STL containers and algorithms
- Vector, list, map, and set containers

#### 6. Templates and Generic Programming

- Introduction to templates and generic programming
- Function templates and class templates
- Template specialization and template metaprogramming
- Benefits and use cases of generic programming

#### 7. Exception Handling

- Handling runtime errors with exception handling
- try, catch, and throw statements
- Exception specifications and standard exception classes
- Best practices for exception handling in C++

#### 8. Introduction to Data Structures

- Overview of data structures and their importance
- Classification of data structures: linear vs. non-linear
- Abstract data types and their implementations

#### 9. Basic Data Structures

- Arrays, linked lists, stacks, and queues
- Implementation and operations of basic data structures
- Applications and use cases of each data structure

#### 10. Advanced Data Structures (Optional)

- Trees, graphs, hash tables, and heaps
- Introduction to advanced data structures
- Implementation and applications of advanced data structures

#### Assessment:

- Regular assignments and quizzes to assess understanding of C++ programming concepts and data structures
- Programming projects to implement and demonstrate proficiency in coding and utilizing data structures
- Mid-term and final examinations covering theoretical concepts and practical applications

Note: This syllabus is subject to modification based on the discretion of the course instructor or institution.

Conclusion: By the end of this course, students will have a solid foundation in C++ programming language and proficiency in implementing and utilizing various data structures. They will be equipped with the skills necessary to analyze problems, design algorithms, and develop efficient software solutions using C++ and data structures. We look forward to guiding you through this learning journey and helping you achieve your programming goals.

Csdtd Centre