[Name of the Writer]

[Name of Instructor]

[Subject]

[Date]

Manual Coursework Outline

1. **Basic Abstract Data Types, Vector, Stack, Queue, Hash, Map, Unordered Map, Set.**

It is a class of operations or a type for objects. The behavior of objects is defined by a set of operations and set of values. In List ADT, get(), size(), replace(), remove(), insert() and isFull() functions can be performed. In Stack ADT pop(), peek(), size(), and push() operations can be performed. In Queue ADT peek(), dequeue(), enqueue() and size() operations can be performed. In ADT vector elementAtRank(), insertAtRank and removeAtRank() are the important operations. The ADT Set stores the unique values without having any specific order. In ADT Map and Hash, a unique key is used to map the values.

1. **Pointers and dynamic memory allocation**

The Dynamic Memory Allocation is used for allocating memory at “run time”. Pointers are used to refer the DMA or Dynamic Memory Allocation.

1. **Inheritance, Polymorphism and visual destructors**

The ability of a lower-level object to inherit or access the data structures and behaviors connected with all the classes. Polymorphism is used to invoke the method determined at the runtime grounded on the object type.

1. **Lambdas**

These are the short, unnamed and inline functions. Lambdas are used to perform operations and functions like reduce () and map (). These operations and functions often considered as one parameter. By creating the function of lambda one can actually organize his or her code.

1. **Range-Based loops**

It is used for executing a loop over a range. It is also used as a readable equivalent to the traditional for the operation of loop over a range of elements. The syntax for Range-Based Loops is:

for ( range \_ declaration : range \_ expression) loop \_ statement

for ( init – statement ( optional ) range \_ declaration : range \_ expression ) loop \_ statement.

1. **Move Semantics**

It is also known as scenic route. It is a novel method of shifting resources everywhere in an optimum mode by evading redundant copies of impermanent objects which are based on the references of rvalue. The syntax is

Class \_ name (class \_ name &&)

Class \_ name (class \_ name &&) = default

Class \_ name (class \_ name &&) = delete

1. **Strongly Typed Enums**

There are many drawbacks of the Enums. The three drawbacks include implicitly conversion to int function. Enums functions are mostly introduced at the end of the body or in enclosing scope. Many limitations and boundaries incurred by regular enums are solved by Strongly Typed Enums.

1. **Uniform Initialization**

It permits the use of consistent syntax to initialize objects and variables varying from primitive type of aggregates. In simpler words it presents brace-initialization that practices braces () to encompass initializer values. The syntax of uniform Initialization is as type var\_nam{arg1, arg2, ….. arg n}..

1. **Constexpr**

The analysis and evaluation of a variable or a function at accumulate time. These expressions can only be used with the constant expressions. It should have specific conditions like it should be in LiteralType and must instantaneously initialize. The body of a function must not had a goto statement, any labeled statement and an asm declaration.

1. **Type Inference**

This term refers to auto detection of the ADT or data type expression used in the programming language. It is a compiler feature of practical programming languages.

1. **C++ Resources**

This is an explanation of most noteworthy functions, classes and objects of the SLA or Standard Language Library with descriptive fully-functional short programs like C++ 11, C++ Language and General C++ Programming.

1. **Modern C++ Resources**

C++ is called Modern C++. It is founded on C++17, C++14 and C++11. The two ultimate resources of Modern C++ are C++ Core Guidelines and C++ Best Practices.