IT CENTER SCHOOL OF TECHNOLOGY: C++ PROGRAMMING MODULE

COURSE TITLE: COMPREHENSIVE INTRODUCTION TO C++ PROGRAMMING

TUTOR:

MR REEKELITSOE NKHABU

The C++ Programming Module at IT Center School of Technology provides a thorough foundation in C++ for beginners. Covering everything from basic syntax and control structures to advanced topics like object-oriented programming and the Standard Template Library (STL), this course prepares students to develop robust and efficient C++ applications. Key learning outcomes include understanding core programming concepts, mastering object-oriented design, and implementing data structures and algorithms.

C++ PROGRAMMING COURSE OUTLINE

COURSE TITLE: COMPREHENSIVE INTRODUCTION TO C++ PROGRAMMING

COURSE DURATION: 12 WEEKS

COURSE DESCRIPTION

This course is designed to provide beginning students with a complete and thorough understanding of C++ programming. It covers fundamental programming concepts, object-oriented programming, data structures, algorithms, and advanced features of C++. By the end of the course, students will be equipped with the skills to develop robust and efficient C++ applications.

COURSE OBJECTIVES

- Understand and apply basic programming concepts in C++.
- Develop proficiency in using control structures and functions.
- Gain a solid foundation in object-oriented programming.
- Learn to manage dynamic memory and use advanced features of C++.
- Implement data structures and algorithms using C++.
- Develop problem-solving and debugging skills.

PREREQUISITES

- Basic understanding of computer operations.
- No prior programming experience is required.

WEEK 1: INTRODUCTION TO C++

TOPICS

- Overview of the Course
- Course objectives and structure
- Introduction to C++
- History and applications of C++
- Setting up the development environment (IDE, Compiler)
- Basic Syntax and Structure
- Writing and running your first C++ program
- Understanding the structure of a C++ program
- Variables and Data Types
- Declaration and initialization
- Fundamental data types (int, float, double, char, bool)
- Input and Output
- Using cin and cout for standard input and output

ASSIGNMENTS

- Install and configure a C++ development environment.
- Write a simple C++ program to display a message.

WEEK 2: CONTROL STRUCTURES

TOPICS

- Conditional Statements
- If, If-Else, and Nested If statements
- Switch-case statement
- Looping Constructs
- For loop
- While loop
- Do-While loop
- Control Flow
- Break and Continue Statements

ASSIGNMENTS

- Write programs using various control
- structures to solve simple problems.

WEEK 3: FUNCTIONS

TOPICS

Function Basics

- Declaration and definition of functions
- Function parameters and return types

Scope and Lifetime of Variable

- Local and global scope

Function Overloading

- Overloading functions with different parameters

Inline Functions

Recursive Functions

ASSIGNMENTS

- Implement programs using different types of functions, including recursive functions.

WEEK 4: ARRAYS AND STRINGS

TOPICS

Arrays

- Declaration and initialization of arrays
- Multidimensional arrays
- Passing arrays to functions

Strings

- C-style strings
- String manipulation functions (strcpy, strlen, etc.)
- Introduction to the C++ string class

ASSIGNMENTS

- Write programs to manipulate arrays and strings.

WEEK 5: POINTERS AND REFERENCES

TOPICS

Pointers

- Pointer basics and pointer arithmetic
- Pointers and arrays
- Dynamic memory allocation with new and delete

References

- Reference variables
- Passing by reference vs. passing by value

ASSIGNMENTS

- Write programs to demonstrate the use of pointers and references.

WEEK 6: OBJECT-ORIENTED PROGRAMMING (OOP) - PART 1

TOPICS

Classes and Objects

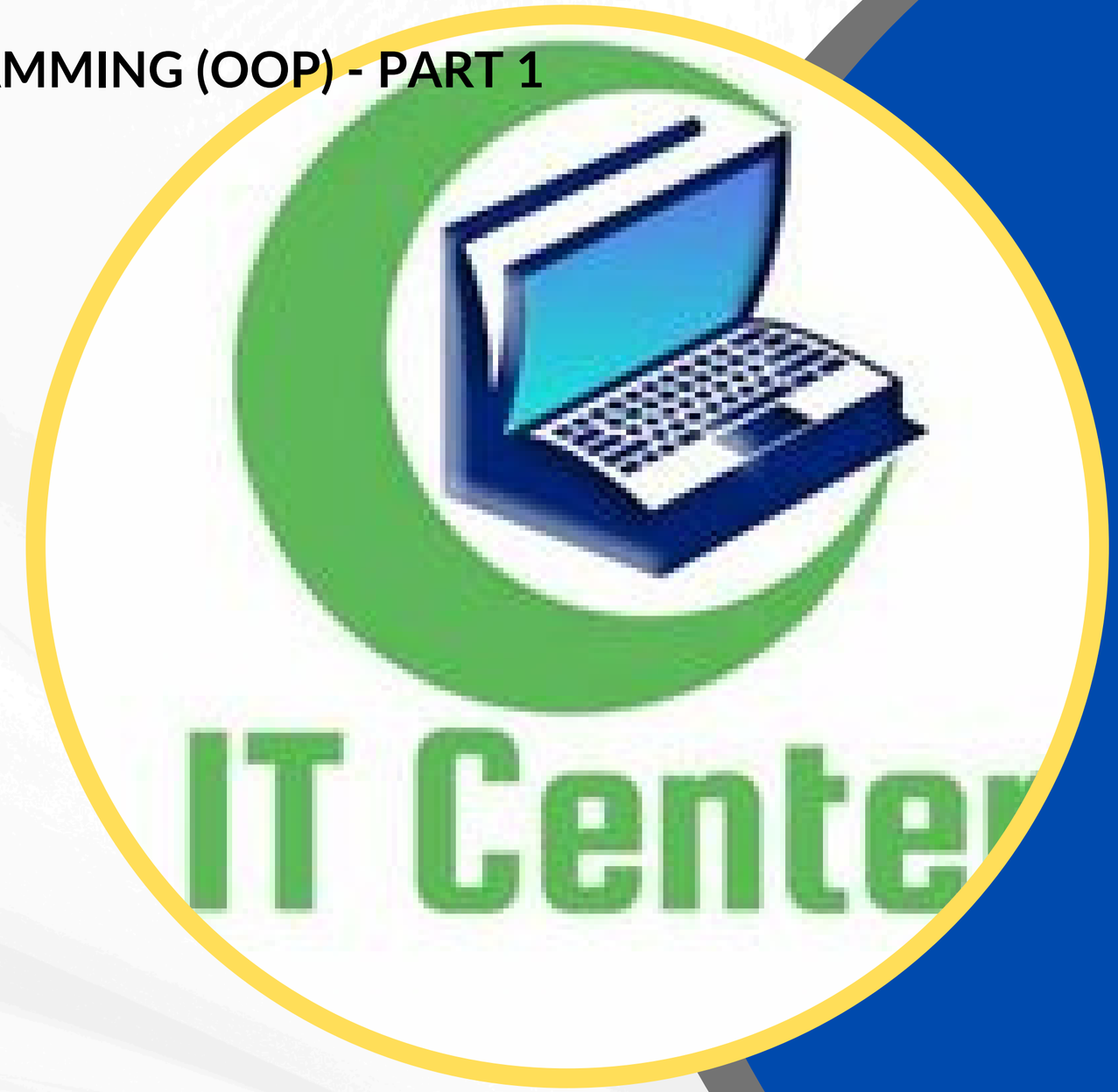
- Defining and creating classes and objects
- Access specifiers: public, private, protected
- Constructors and destructors

Member Functions

- Defining and accessing member functions
- Inline member functions

ASSIGNMENTS

- Implement basic classes with constructors and member functions.



WEEK 7: OBJECT-ORIENTED PROGRAMMING (OOP) - PART 2

TOPICS

Advanced OOP Concepts

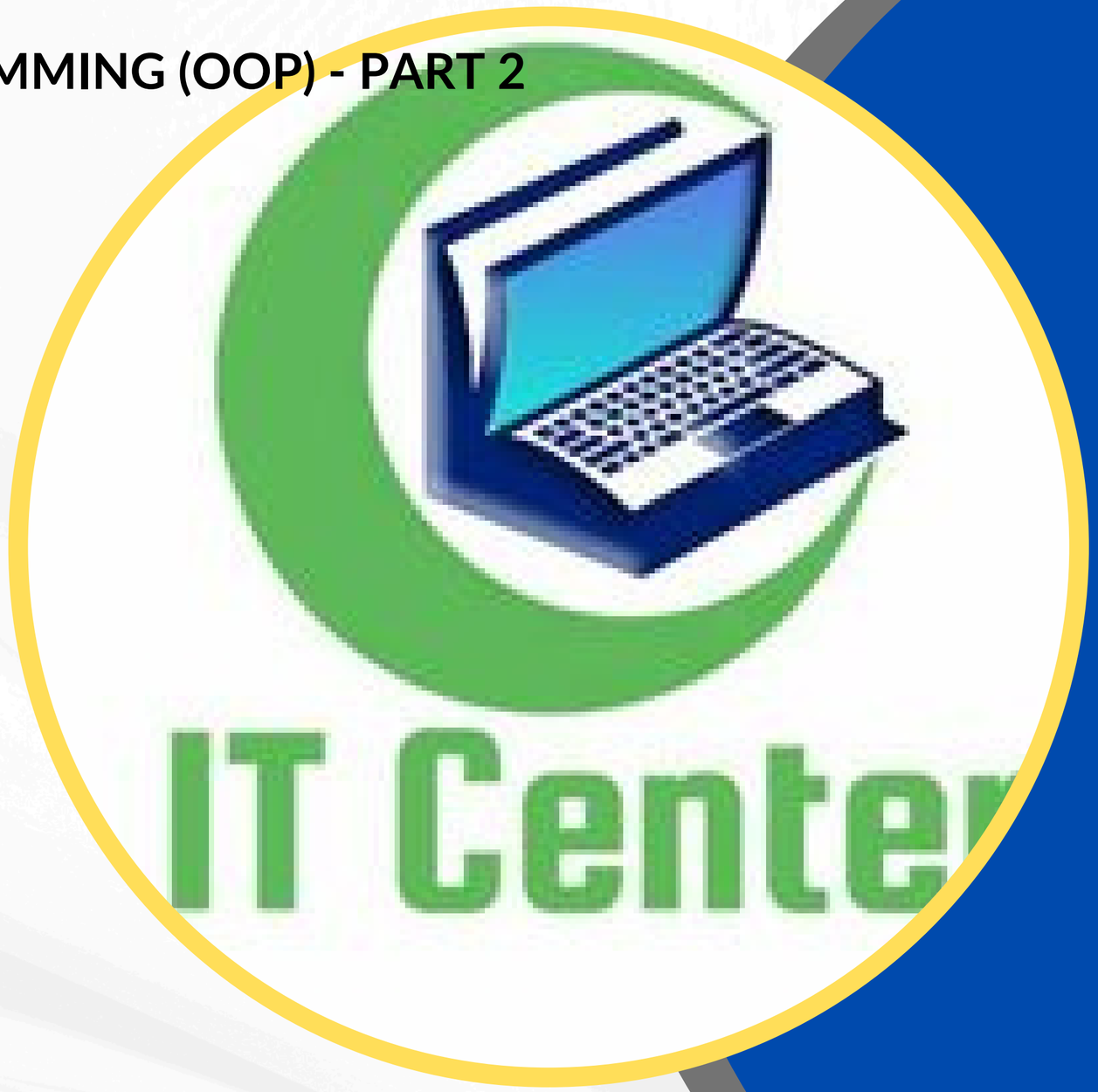
- Static members and functions
- Friend functions and classes

Inheritance

- Basics of inheritance
- Types of inheritance (single, multiple, hierarchical)
- Function overriding

ASSIGNMENTS

- Write programs to demonstrate inheritance and use of static members.



WEEK 8: OBJECT-ORIENTED PROGRAMMING (OOP) - PART 3

TOPICS

Polymorphism

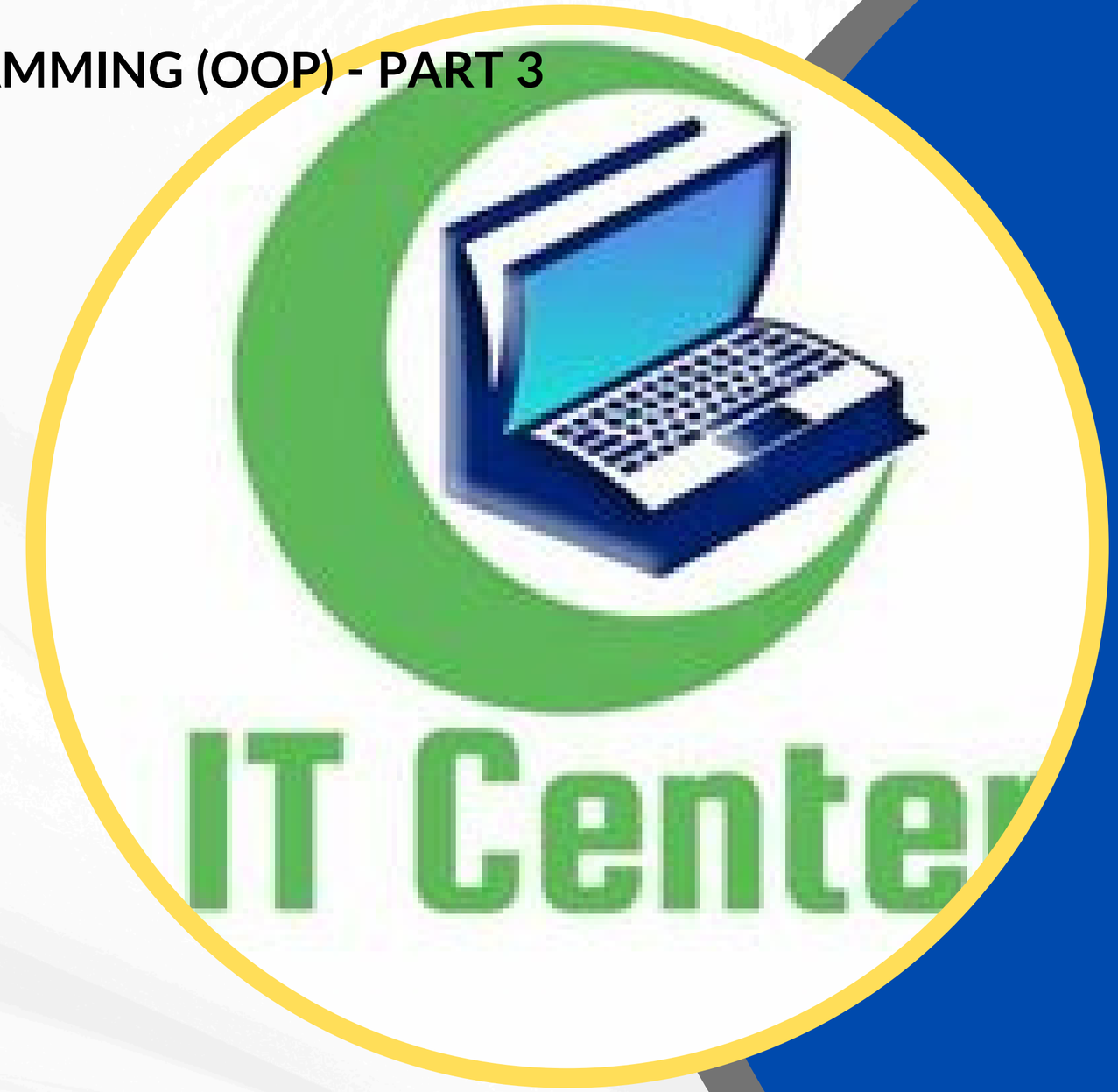
- Pointers to derived classes
- Virtual functions and dynamic binding
- Abstract classes and pure virtual functions

Operator Overloading

- Overloading operators (e.g., +, -, *, /, ==, <<, >>)
- Overloading unary and binary operators

ASSIGNMENTS

- Implement polymorphism and operator overloading in C++ programs.



WEEK 9: TEMPLATES AND THE STANDARD TEMPLATE LIBRARY (STL)

TOPICS

Templates

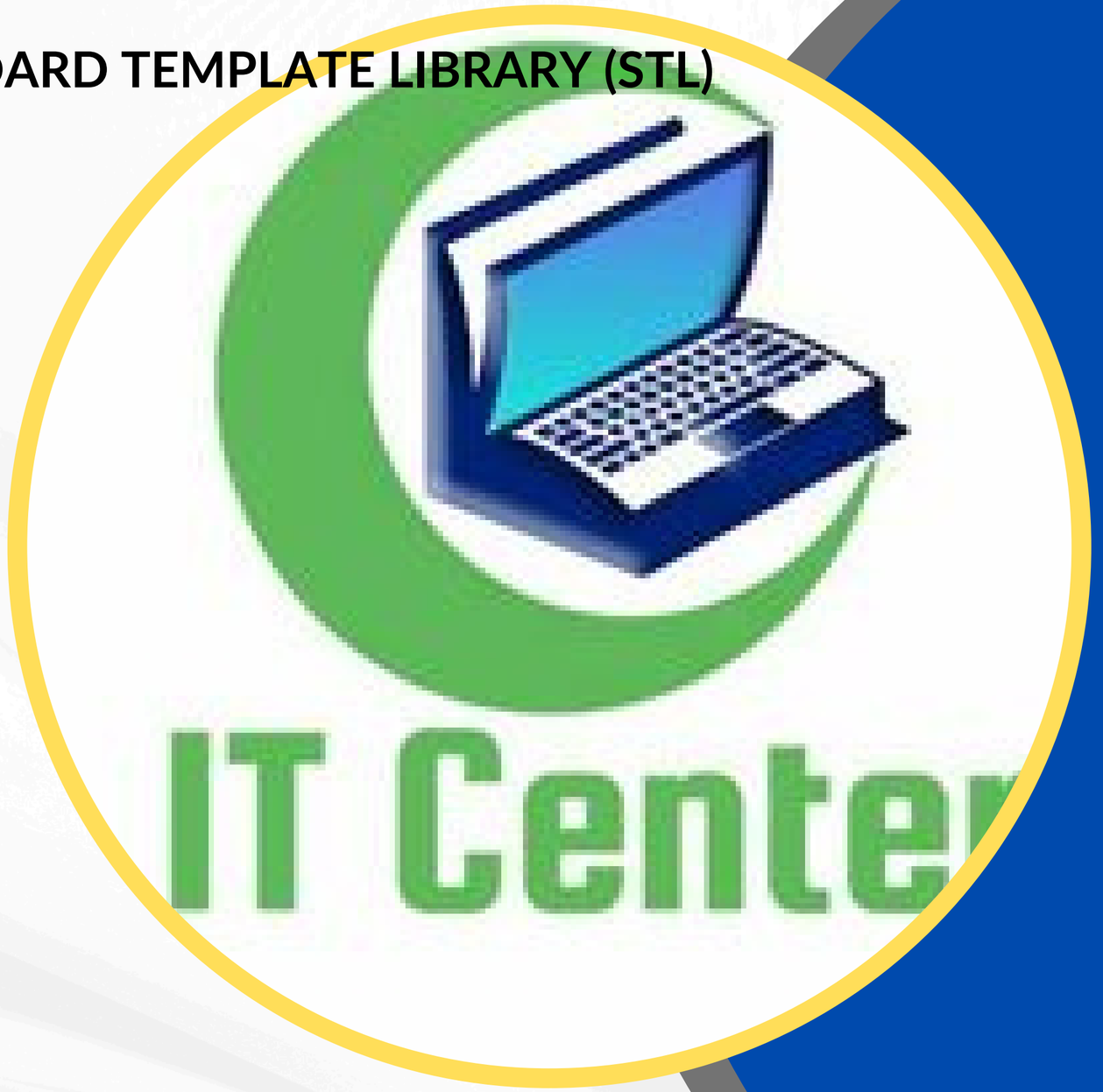
- Function templates
- Class templates

Standard Template Library (STL)

- Introduction to STL
- Containers (vector, list, map, set)
- Iterators
- Algorithms (sort, find, etc.)

ASSIGNMENTS

- Write programs using templates and STL containers and algorithms.



WEEK 10: ADVANCED TOPICS

TOPICS

Exception Handling

- Try, catch, and throw statements
- Creating and handling custom exceptions

File I/O

- File streams (ifstream, ofstream, fstream)
- Reading from and writing to files

Lambda Expressions and Modern C++

- Introduction to lambda expressions
- Overview of modern C++ features (C++11/14/17/20)

ASSIGNMENTS

- Implement exception handling and file I/O operations in C++ programs.

WEEK 11: DATA STRUCTURES AND ALGORITHMS

TOPICS

Basic Data Structures

- Arrays, linked lists, stacks, queues
- Trees and graphs

Algorithms

- Sorting algorithms (bubble sort, quicksort, merge sort)
- Searching algorithms (linear search, binary search)

Complexity Analysis

- Time and space complexity
- Big O notation

ASSIGNMENTS

Implement basic data structures and algorithms in C++.

WEEK 12: FINAL PROJECT AND REVIEW

TOPICS

Final Project

- Project guidelines and expectations
- Development, debugging, and testing support

Course Review

- Recap of key concepts and topics
- Q&A session

Final Assessment

- Final exam covering the entire course

ASSIGNMENTS

- Complete a final project demonstrating the application of course concepts.

ABOUT
OUR VISION

The IT Center School of Technology is a premier educational institution dedicated to fostering technical expertise and innovation in the field of information technology. With a strong emphasis on practical learning, our programs are designed to equip students with the skills and knowledge needed to excel in the ever-evolving tech industry. Our experienced instructors, state-of-the-art facilities, and comprehensive curriculum ensure that students receive a high-quality education that prepares them for real-world challenges.

The IT Center School of Technology is a premier educational institution dedicated to fostering technical expertise and innovation in the field of information technology. With a strong emphasis on practical learning, our programs are designed to equip students with the skills and knowledge needed to excel in the ever-evolving tech industry. Our experienced instructors, state-of-the-art facilities, and comprehensive curriculum ensure that students receive a high-quality education that prepares them for real-world challenges.

THANK YOU