Object-Oriented Programming Lab – Lab 2

Lab Title:
Exception Handling and Templates in C++

Duration:
2 Hours

Total Questions:
3

Topics Covered:
• Exception Handling
• Function Templates
• Class Templates

Objective:
- To understand the concept of exception handling in C++.
- To implement function and class templates for code reusability.
- To gain hands-on experience with C++ error-handling mechanisms and generic programming.

Question 1: Exception Handling – Division by Zero
Problem Statement:
Write a program that accepts two numbers and performs division. Use exception handling to manage the case when the denominator is zero. The program should display an appropriate message when an exception is caught.

Expected Competency:
- Use of try-catch blocks
- Throwing and catching exceptions
- Proper user input handling

Time Allocation: 30 minutes

Question 2: Function Template – Find Maximum of Two Values

Problem Statement:
Create a function template that accepts two parameters of any data type and returns the maximum value. Demonstrate the function with various data types (e.g., int, float, char).

Expected Competency:
- Defining and using function templates
- Understanding generic programming
- Template instantiation with multiple data types

Time Allocation: 40 minutes

Question 3: Class Template – Stack Implementation

Problem Statement:
Design a class template for a stack that supports the following operations:
- Push an element onto the stack
- Pop an element from the stack
- Display all elements in the stack

Demonstrate the stack with at least one data type (e.g., int).

Expected Competency:
- Creating and using class templates
- Managing stack operations (LIFO)
- Applying templates in user-defined classes

Time Allocation: 50 minutes

Viva Questions (Sample):

1. What is an exception? How is it different from an error?
2. Explain the role of try, catch, and throw in C++.
3. What are templates? Why are they useful in programming?
4. Differentiate between function templates and class templates.
5. Can we have multiple catch blocks in C++? If so, why?