National University of Computer and Emerging Sciences

Laboratory Manual

for

Object Oriented Programming Lab

Course Instructor	Mr. Uzair Naqvi
Lab Instructor(s)	Aqib Zeeshan, Seemab Ayub
Section	BCS-2E
Date	Wednesday, 8 May 2024
Semester	Spring 2024

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

Objectives:

In this lab students will practice:

- Exception Handling

1. Exercise: Marks: 05

Write a program that throws and catches an integer exception. Handle the exception and print its value.

- Update the program so that it can handle both integer and double exceptions (add another catch block for double).
- Add another catch(...) block to handle any other exception.
- Write the main function, use if or switch statements to enter different data types.

2. Exercise: Marks: 05

Create a method add(double a, double b). If either argument is non-numeric, throw a std::invalid_argument exception (built in c++ exception). Throw and catch this exception in main and print the error message.
You can use the std::isnan() function (defined in cmath library) to check if an argument is not a number.

3. Exercise: Marks: 10

Create a C++ program that demonstrates exception handling for array index out of bound scenarios. Implement custom exception handling to catch and handle the situation where the user tries to access an element beyond the bounds of the array.

Your program should include the following:

- Define an integer array of size 5.
- Implement a function named getElementAtIndex(int index) that takes an index as input and returns the value at that index in the array.
- Inside the function, implement custom exception handling to catch the array index out of bounds scenario. If the index is out of bounds (less than 0 or greater than 4), throw a custom exception of type ArrayIndexOutOfBoundsException.
- For custom exception classes, create an exception class (ArrayIndexOutOfBoundsException) and inherit it from the c ++ exception class.
- Add a printMessage method to the exception class.
- In the main function, prompt the user to input an index and display the value at that index in the array using the getElementAtIndex function. Handle any potential exceptions and display appropriate error messages.

4. Exercise: Marks: 10

Create a C++ program that implements a Fraction class to represent fractions. Implement exception handling to handle division errors such as division by zero.

- Implement a class named Fraction with private data members numerator and denominator.
- Include a constructor that initializes the fraction with numerator and denominator.
- Implement a method named divide(const Fraction& other) that divides the current fraction by another fraction.
- Inside the divide method, handle the division by zero scenario by throwing a custom exception DivisionByZeroException if the denominator of the other fraction is zero.
- In the main function, create two Fraction objects and attempt to divide one by the other using the divide method. Handle any potential exceptions and display appropriate error messages.