

## **Assignment Part 5**

44. Two integers are taken from keyboard. Then perform division operation. Write a try block to throw an exception when division by zero occurs and appropriate catch block to handle the exception thrown.
45. Write a C++ program to demonstrate the use of try, catch block with the argument as an integer and string using multiple catch blocks.
46. Create a class with member functions that throw exceptions. Within this class, make a nested class to use as an exception object. It takes a single const char\* as its argument; this represents a description string. Create a member function that throws this exception. (State this in the function's exception specification.) Write a try block that calls this function and a catch clause that handles the exception by displaying its description string.
47. Design a class Stack with necessary exception handling.
48. Write a Garage class that has a Car that is having troubles with its Motor. Use a function-level try block in the Garage class constructor to catch an exception (thrown from the Motor class) when its Car object is initialized. Throw a different exception from the body of the Garage constructor's handler and catch it in main().
49. Vehicles may be either stopped or running in a lane. If two vehicles are running in opposite direction in a single lane there is a chance of collision. Write a C++ program using exception handling to avoid collisions. You are free to make necessary assumptions.
50. Write a template function max() that is capable of finding maximum of two things (that can be compared). Used this function to find (i) maximum of two integers, (ii) maximum of two complex numbers (previous code may be reused). Now write a specialized template function for strings (i.e. char\*). Also find the maximum of two strings using this template function.
51. Write a template function swap() that is capable of interchanging the values of two variables. Used this function to swap (i) two integers, (ii) two complex numbers (previous code may be reused). Now write a specialized template function for the class Stack (previous code may be reused). Also swap two stacks using this template function.
52. Create a C++ template class for implementation of Stack data structure. Create a Stack of integers and a Stack of complex numbers created earlier (code may be reused). Perform some push and pop operations on these stacks. Finally print the elements remained in those stacks.