

Nithin S
221IT085

IT150 Class Assignment

1. Write a C++ program to check the duplicate mobile number by throwing a custom exception.

The class named ContactDetail with the following private member variables:

Data type Variable name string mobilenumber
string alternateMobilenumber string email

In ContactDetail class include the following member function:

Member Function Description void display() : This function is used to display the contact details

Create a class named ContactDetailBO with the following member function

Method	name	Description
bool	validate(ContactDetail cd) :	

This method is used to validate whether the mobile number and alternate mobile number are not the same. If both are the same, then it

throws "DuplicateMobileNumber Exception". Otherwise, return true to print the details.

Class named DuplicateMobileNumberException that extends exception class and create a method to return the following message "DuplicateMobileNumberException".

CODE

```
#include <iostream>
#include <stdexcept>
#include <string>

class DuplicateMobileNumberException : public std::exception {
public:
    const char* what() const noexcept override {
        return "DuplicateMobileNumberException";
    }
};

class ContactDetail {
private:
    std::string mobilenumber;
    std::string alternateMobilenumber;
    std::string email;

public:
    ContactDetail(const std::string& mobile, const std::string& alternate, const std::string& mail) :
        mobilenumber(mobile), alternateMobilenumber(alternate), email(mail) {}

    void display() {
        std::cout << "Mobile Number: " << mobilenumber << std::endl;
        std::cout << "Alternate Mobile Number: " << alternateMobilenumber << std::endl;
        std::cout << "Email: " << email << std::endl;
    }

    friend class ContactDetailBO;
};

class ContactDetailBO {
public:
    bool validate(const ContactDetail& cd) {
        if (cd.mobilenumber == cd.alternateMobilenumber)
            throw DuplicateMobileNumberException();
        return true;
    }
};
```

```

int main() {
    std::string mobile = "1234567890";
    std::string alternate = "0987654321";
    std::string email = "example@example.com";

    ContactDetail contact(mobile, alternate, email);
    ContactDetailBO bo;

    try {
        if (bo.validate(contact)) {
            std::cout << "Contact details are valid." << std::endl;
            contact.display();
        }
    } catch (const DuplicateMobileNumberException& e) {
        std::cerr << "Exception caught: " << e.what() << std::endl;
    }

    return 0;
}

```

Output

```

nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_7$ g++ main.cpp
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_7$ ./a.out
Contact details are valid.
Mobile Number: 1234567890
Alternate Mobile Number: 0987654321
Email: example@example.com
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_7$ |

```

OUTPUT

```
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_6$ g++ 1.cpp
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_6$ ./a.out
Enter the numerator: 6
Enter the denominator: 0
Exception caught: Division by zero exception
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_6$ |
```