OOP Lab Assignment 4

Jaynam Modi. PG-43. G3. Sep 06, 2020

- Problem Statement : Define a class Employee consisting following :

  - Data members :

    1. Employee ID
    2. Name of Employee
    3. Age
    4. Income
    5. City
    6. Vehicle

  - Member Functions :

    - To assign initial values.
    - To display.

Accept Employee ID, Name, Age, Income, City and Vehicle from the user. Create an exception to check the following conditions and throw an exception if the condition does not meet.

- Employee age between 18 and 55
- Employee income between Rs. 50,000 - Rs. 1,00,000 per month
- Employee staying in Pune/ Mumbai/ Bangalore / Chennai
- Employee having 4-wheeler

- Objectives :

1. To learn the concepts of Exception handling in C++.
2. To learn and use exception handling mechanism using try catch block

- Theory :

When executing C++ code, different errors can occur : coding errors made by the programmer,

errors due to wrong input, or other unforeseeable things.

When an error occurs, C++ will normally stop and generate an error message. The technical term for this is : C++ will throw an exception (throw an error).

Exception handling is the process of handling errors and exceptions in such a way that they do not hinder normal execution of the system.

Exception handling in C++ consist of three keywords : try, throw and catch :

The try statement allows you to define a block of code to be tested for errors while it is being executed.

The throw keyword throws an exception when a problem is detected, which lets us create a custom error.

The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

If you do not know the throw type used in the try block, you can use the "three dots" syntax (...) inside the catch block, which will

handle any type of exception

If a catch block cannot handle the particular exception it has caught, you can rethrow the exception. The rethrow expression causes the originally thrown object to be rethrown.

- Algorithm :

1. Initialise.
2. Take input from user.
3. Verify Integrity of Input using the given rules and Exception Handling.
4. Provide Results accordingly.

- Platform :

  - 64-bit Open source Linux or its derivatives.
  - Open Source C++ Programming tool like G++/Eclipse Editor.

- Input : Employee ID, Name, Age, Income, City and Vehicle of an Employee.

- Output : Employee Information if no exception or Exception Messages if there is an Exception.

- Conclusion : Thus, use the exception handling mechanism to display information of an Employee.

- FAQs :

1. Why do we use Exception Handling mechanism?

> Exception handling is the process of handling errors and exceptions in such a way that they do not hinder normal execution of the system.

2. Is it possible to use multiple catch for single throw? Explain?

> A single try statement can have multiple catch statements. Execution of particular catch block depends on the type of exception thrown by the throw keyword. If throw keyword send exception of integer type, catch block with integer parameter will get execute.

3. What is Exception Specification?

> Exception specifications are a C++ language feature that indicate the programmer's intent about the exception types that can be propagated by a function. You can specify that a function may or may not exit by an exception by using an exception specification.

4. What is Re-throwing Exception?

> If a catch block cannot handle the particular exception it has caught, you can rethrow the exception. The rethrow expression causes the originally thrown object to be rethrown.

5. What is stack unwinding?

> In C++, when an exception occurs, the function call stack is linearly searched for the exception handler, and all the entries before the function with exception handler are removed from the function call stack.

```cpp
#include <iostream>

using namespace std;

class Employee {
    public:
        int id;
        string name;
        int age;
        double income;
        string city;
        string vehicle;

        void validate(){
            try{
                if (age < 18 or age > 55){
                    throw 1855;
                }

                if (income < 50000 or income > 100000){
                    throw 50100;
                }

                if (city != "Pune" && city != "Bangalore"
&& city != "Mumbai" && city != "Chennai"){
                    throw 400;
                }

                if (vehicle != "4-Wheeler"){
                    throw 404;
                }
            } catch (int exc) {
                if (exc == 1855){
                    cout << " > Age not valid." << endl;
                } else if (exc == 50100){
                    cout << " > Income Invalid." << endl;
                } else if (exc == 400){
                    cout << " > City Invalid." << endl;
                } else if (exc == 404){
                    cout << "Vehicle Invalid." << endl;
                }
            } catch (...) {
                cout << " > UNKNOWN ERROR." << endl;
            }
        }

        Employee(){
            cout << "Enter the Following Details \n";
      cout << "       Employee ID : ";
      cin >> id;
      cout << "       Name : ";
      cin.ignore();
      getline(cin, name);
      cout << "       Age : ";
      cin >> age;
      cout << "       Income : ";
      cin >> income;
      cout << "       City : ";
      cin.ignore();
      getline(cin, city);
      cout << "       Vehicle : ";
      cin.ignore();
      getline(cin, vehicle);

      validate();
        }

        Employee(int a, string b, int c, double d, string e,
string f){
            id = a;
            name = b;
            age = c;
            income = d;
            city = e;
            vehicle = f;

            validate();
        }

        void print(){
            cout << "Entered Details are : " << endl;
      cout << "       Employee ID : " << id << endl;
      cout << "       Name : " << name << endl;
      cout << "       Age : " << age << endl;
      cout << "       Income : " << income << endl;
      cout << "       City : " << city << endl;
      cout << "       Vehicle : " << vehicle << endl;
        }

};


int main(){
    Employee e1;
    e1.print();
}
```

```
u0_a362@localhost:~$ ./a.out
Enter the Following Details
        Employee ID : 1244
        Name : Marcus Holloway
        Age : 17
        Income : 70000
        City : Pune
        Vehicle : 4-Wheeler
 > Age not valid.
Enter the Following Details
        Employee ID : 1244
        Name : Marcus Holloway
        Age : 19
        Income : 45000
        City : Pune
        Vehicle : 4-Wheeler
 > Income Invalid.
```