LASALLE COLLEGE VANCOUVER

Error Handling. Exception

VGP 131 - Object Oriented Programming in C++ II

Instructor: Ivaldo Tributino

May 30, 2022

ASSIGNMENT INSTRUCTION

- The exam must be submitted by Mar 05, 2022.
- Each problem presents its own score, the sum of all scores is 100.

Student's Number:
Student's Name:

(15 POINTS) PROBLEM 1

Create a program that loops from 1 to 50 and throws an exception when it finds a value divisible by 5 and 7.

(15 POINTS) PROBLEM 2

Consider the following example:

```
class Base
{
public:
    Base() {}
    virtual void print() { std::cout << "Base" << endl; }
};

class Derived: public Base
{
public:
    Derived() {}
    void print() override { std::cout << "Derived"<< endl; }
};
```

```
int main()
    try
    {
        try
        {
            throw Derived{};
        }
        catch (Base& b)
        {
            std::cout << "Caught␣Base␣b,␣which␣is␣actually␣a␣";
            b.print();
            throw b;
        }
    }
    catch (Base& b)
    {
        std::cout << "Caught␣Base␣b,␣which␣is␣actually␣a␣";
        b.print();
    }
```

In the above example we throw an exception of type Derived. However, the output of this program is:
Caught Base b, which is actually a Derived
Caught Base b, which is actually a Base
Explain what happened.

(15 POINTS) PROBLEM 3

Using the code from the previous problem. How to provide a way to rethrow the exact same exception that was just caught?

(15 POINTS) PROBLEM 4

Create your own exception class for throwing and catching (inherit from std::exception).

(20 POINTS) PROBLEM 5

Complete the following class to get the desired output.

```
class Rectangle
{
protected:
    double _length;
    double _width;
public:
```

```
    double area(){
        return _length*_width;
    }
    ~ Rectangle(){ }
};
```

```
try
{
    Rectangle();
    Rectangle(-1,3);
}
catch (const std::exception& ex)
{
    cout << ex.what() << endl;
}
// The expected output:
Rectangle created
Rectangle destroyed
From Base: Data members must be values greater than 0.0
```

(20 POINTS) PROBLEM 6

Complete the subclass RectangularPrism which inherits the above class and will throw exceptions as shown in the examples below.

```
class RectangularPrism : public Rectangle
{
private:
    double _higth;

public:


    double surfaceArea(){
        return 2*(_length*_width + _width*_higth + _length*_higth );
    }

    double volume(){
        return Rectangle::area()*_higth;
    }

    ~RectangularPrism(){ }

};
```

```
try
{
    RectangularPrism(-1,2,3);
}
catch (const std::exception& ex)
{
    cout << ex.what() << endl;
}
try
{
    RectangularPrism(1,2,-3);
}
catch (const std::exception& ex)
{
    cout << ex.what() << endl;
}

Output:
From Base: Data members must be values greater than 0.0
Rectangle created
Rectangle destroyed
From Derived: height must be greater than 0.0
```