## Question –

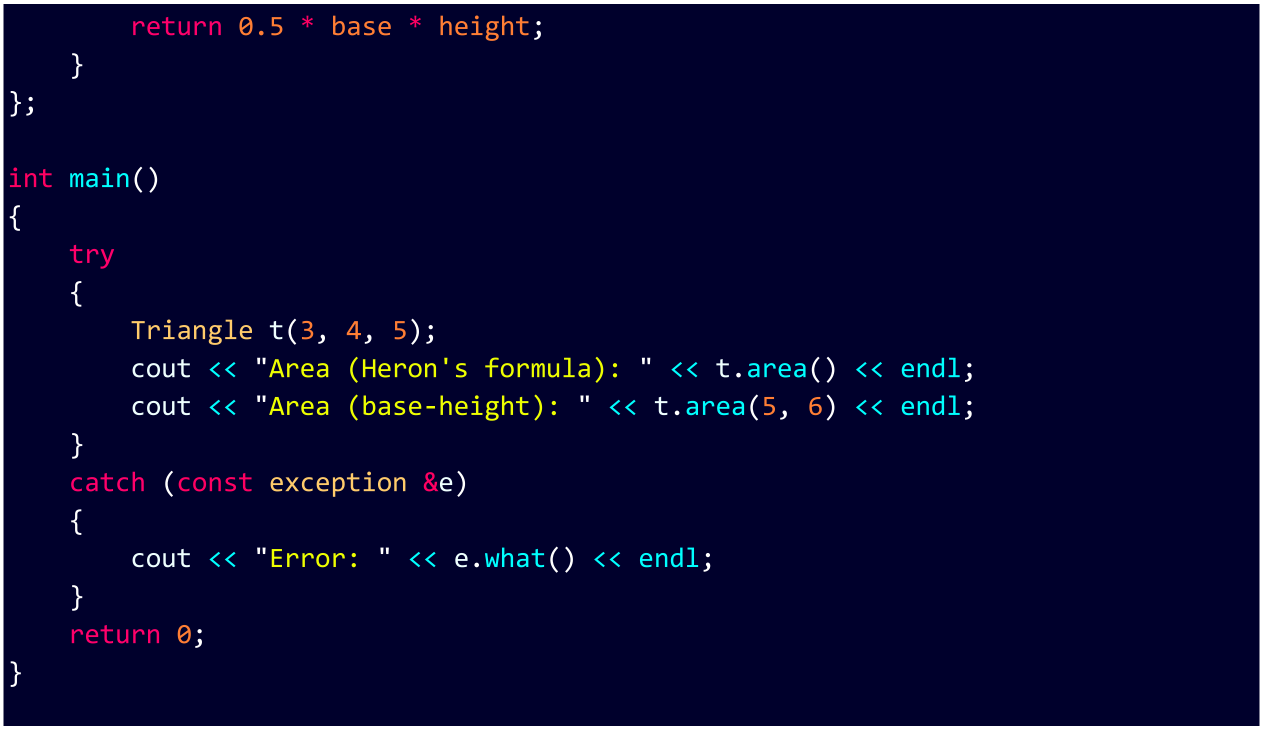
Create a triangle class and add exception handling statements to ensure the following conditions:

1. Side should be greater than 0,
2. Sum of 2 sides should be greater than 3rd side.

Also find the area using overloaded functions.

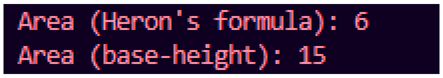
## Code –

```
#include <iostream>
#include <cmath>
#include <stdexcept>
using namespace std;

class Triangle
{
    double a, b, c;

public:
    Triangle(double x, double y, double z)
    {
        if (x <= 0 || y <= 0 || z <= 0)
            throw invalid_argument("All sides must be greater than 0");
        if (x + y <= z || x + z <= y || y + z <= x)
            throw invalid_argument("Sum of any two sides must be greater than the
third side");
        a = x;
        b = y;
        c = z;
    }

    double area()
    {
        double s = (a + b + c) / 2.0;
        return sqrt(s * (s - a) * (s - b) * (s - c));
    }

    double area(double base, double height)
    {
        if (base <= 0 || height <= 0)
            throw invalid_argument("Base and height must be greater than 0");
```



```cpp
        return 0.5 * base * height;
    }
};

int main()
{
    try
    {
        Triangle t(3, 4, 5);
        cout << "Area (Heron's formula): " << t.area() << endl;
        cout << "Area (base-height): " << t.area(5, 6) << endl;
    }
    catch (const exception &e)
    {
        cout << "Error: " << e.what() << endl;
    }
    return 0;
}
```

## Output –



```
Area (Heron's formula): 6
Area (base-height): 15
```