"Heaven's Light is Our Guide"

Rajshahi University of Engineering & Technology, Rajshahi

Department of Electrical & Computer Engineering
(ECE-20)

Course Code: 1204
Course Title: Object Oriented Programming Sessional
Assignment: Learning Inheritance using Java & C++

Date of Submission: 7-11-2022

Submitted To
Rakibul Hassan
Assistant Professor
Dept. Of Electrical & Computer
Engineering

Submitted By
Name: Md. Tajim An Noor
Roll: 2010025
Session: 2020-21

Objective: To check inheritance in Object Oriented Programing.

Methodology: When a class derives from another class, it's called inheritance ion object oriented programming. The child class will inherit all the public and protected properties and methods from the parent class. In addition, it can have its own properties and methods.
In java, the "extend" keyword is used to implement this concept. In C++, simply mentioning the base class after ":" is used.
To check this concept is OOP, we used a base class called GeometryShape. The inherited classes from this will be Triangle and Quadrangle. And to check multi-level inheritance, more classes was inherited from the Quadrangle class, named Square and Rectangle. To check multiple inheritance, that only works in C++, another class was inherited from Rectangle and Square.
To simply check if a function in a base class can be accessed from an object of an inherited class, some print line statement was used. For multiple inheritance case in C++, there's some cases where ambiguity arises while accessing methods of a base class. To remove this, the virtual keyword is used.
In this experiment, all sorts of inertance will be implemented: Single, multilevel, hierarchical, hybrid, multiple.

Code Implementation:
Java: Single, Multilevel, Hierarchical, Hybrid
Main with base class:

```
package Labs.InheritanceTest.Java;

public class GeometricShape {
    private String name = "Inheritence";
    protected String centre = "I am here";

    protected void printMain() {
        System.out.println(this.name);
    }

    public static void main(String[] args) {
        Triangle tri1 = new Triangle();
        Rectangle rect1 = new Rectangle();
        Square sqr1 = new Square();
        tri1.printCheck();
        rect1.printSmall();
        sqr1.printCheck();
    }

}
```

Triangle Class:

```
package Labs.InheritanceTest.Java;

public class Triangle extends GeometricShape {
    private String name = "Triangle";
    protected String largestAngle = "120";
    protected String totalSides = "There's three sides";

    public void printCheck() {
        printMain();
        System.out.println(this.name);
        System.out.println(largestAngle);
        System.out.println(totalSides);
        System.out.println(centre);
        System.out.println();
    }
}
```

Quadrangle class:

```
package Labs.InheritanceTest.Java;

public class Quadrangle extends GeometricShape {
    private String name = "Quadrangles";
    protected String largestAngle = "110";
    protected String totalSides = "There's four sides";

    public void printCheck() {
        System.out.println(this.name);
        System.out.println(largestAngle);
        System.out.println(totalSides);
        System.out.println(centre);
        System.out.println();
    }
}
```

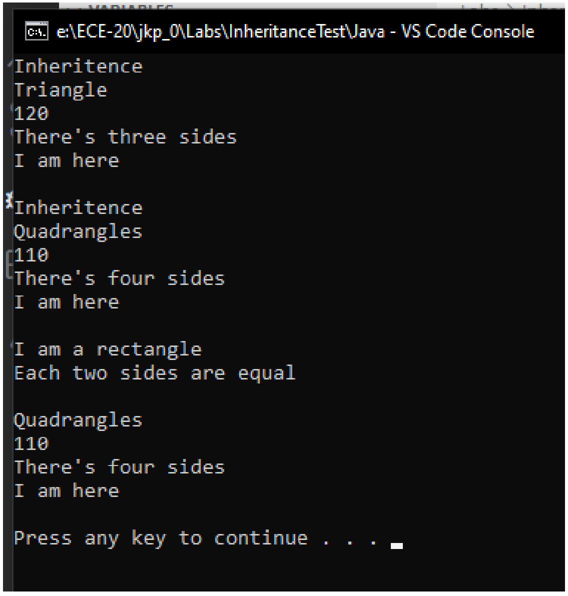
Rectangle Class:

```
package Labs.InheritanceTest.Java;

public class Rectangle extends Quadrangle {
    private String name = "I am a rectangle";
    private String sideSize = "Each two sides are equal";

    public void printSmall() {
        printMain();
        printCheck();
        System.out.println(this.name);
        System.out.println(sideSize);
        System.out.println();
    }
}
```

Square Class:

```
package Labs.InheritanceTest.Java;

public class Square extends Quadrangle {
    private String name = "I am a square";
    private String sideSize = "All sides are equal";
    public void printSmall() {
        printMain();
        printCheck();
        System.out.println(this.name);
        System.out.println(sideSize);
        System.out.println();
    }
}
```

Output:



e:\ECE-20\jkp_0\Labs\InheritanceTest\Java - VS Code Console

```
Inheritence
Triangle
120
There's three sides
I am here

Inheritence
Quadrangles
110
There's four sides
I am here

I am a rectangle
Each two sides are equal

Quadrangles
110
There's four sides
I am here

Press any key to continue . . .
```

C++:

Main:

```
#include <bits/stdc++.h>
#include "GeoShape.hpp"
#include "Quadrangle.hpp"
#include "Triangle.hpp"
#include "Square.hpp"
#include "Rectangle.hpp"
#include "SmallSqr.hpp"
using namespace std;
int main()
{
    Triangle tri1;
    Rectangle rect1;
    Square sqr1;
    SmallSqr smlsq;
    tri1.printCheck();
    rect1.printSmall();
    sqr1.printCheck();
    smlsq.smallPrint();
}
```

Triangle Class:

```
class Triangle : public GeoShape
{
private:
    string name = "Triangle";

protected:
    string largestAngle = "120";

protected:
    string totalSides = "There's three sides";

public:
    void printCheck()
    {
        printMain();
        cout << name << endl;
        cout << largestAngle << endl;
        cout << totalSides << endl;
        cout << centre << endl
             << endl;
    }
};
```

Quadrangle Class:

```
using namespace std;
class Quadrangle : public GeoShape
{
private:
    string name = "Quadrangles";

protected:
    string largestAngle = "110";

protected:
    string totalSides = "There's four sides";

public:
    void printCheck()
    {
        cout << name << endl;
        cout << largestAngle << endl;

        cout << totalSides << endl;
        cout << centre << endl
             << endl;
    }
};
```

Rectangle Class:

```
class Rectangle : public virtual Quadrangle
{
private:
    string name = "I am a rectangle";

private:
    string sideSize = "Each two sides are equal";

public:
    void printSmall()
    {
        printMain();
        printCheck();
        cout << name << endl;
        cout << sideSize << endl
             << endl;
    }
};
```

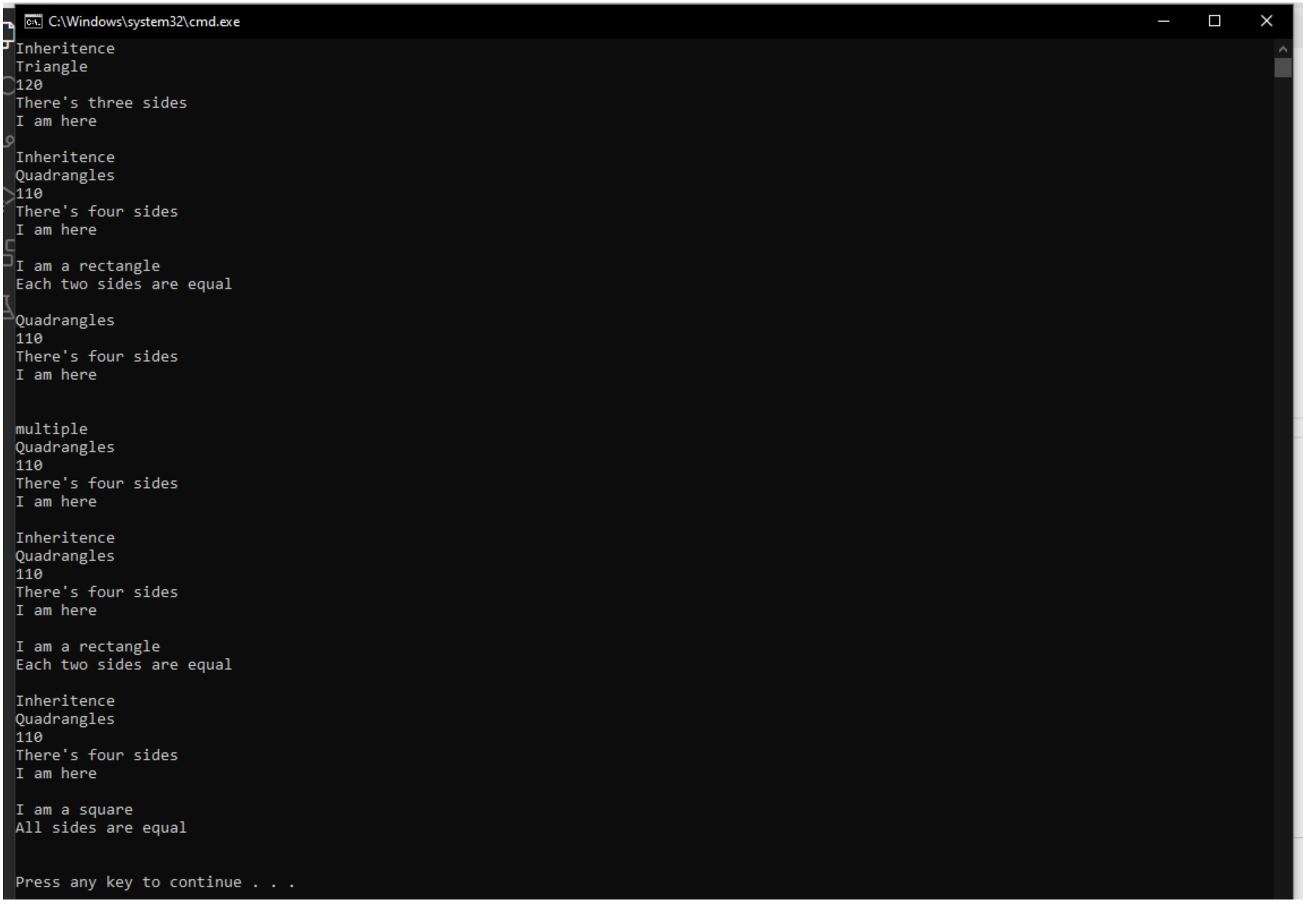
Square Class:

```
class Square : public virtual Quadrangle
{
private:
    string name = "I am a square";

private:
    string sideSize = "All sides are equal";

public:
    void printSSmall()
    {
        printMain();
        printCheck();
        cout << name << endl;
        cout << sideSize << endl
             << endl;
    }
};
```

SmallSquare Class (Multiple inhetance):

```
class SmallSqr : public Rectangle, public Square
{
public:
    void smallPrint()
    {
        cout << endl
             << "multiple" << endl;
        printCheck();
        printSmall();
        printSSmall();
    }
};
```

Output:



```
Inheritence
Triangle
120
There's three sides
I am here

Inheritence
Quadrangles
110
There's four sides
I am here

I am a rectangle
Each two sides are equal

Quadrangles
110
There's four sides
I am here


multiple
Quadrangles
110
There's four sides
I am here

Inheritence
Quadrangles
110
There's four sides
I am here

I am a rectangle
Each two sides are equal

Inheritence
Quadrangles
110
There's four sides
I am here

I am a square
All sides are equal


Press any key to continue . . .
```