



Department Of Electrical Engineering and Computer Sciences

Instructor: Mehreen Tahir

Date: 27th November 2023

Lab Engineer: Mehwish Kiran

Time: 10:00am – 12:50pm

CS 212: Object Oriented Programming

Lab 10: Function Overriding

Information	Description
Name:	Irfa Farooq
CMS ID:	412564
Class:	BEE-14
Section:	D
Tenure:	Fall 2023



Task 1: Implement class shape hierarchy using inheritance.

Shape.h:

```
#pragma once
#include <iostream>

class Shape {
private:
    double length;
    double width;
public:
    Shape();
    void set_length(double);
    double get_length();
    void set_width(double);
    double get_width();
};
```

Rectangle.h:

```
#pragma once
#include <iostream>
#include "Shape.h"

class Rectangle:public Shape {
public:
    void Area();
};
```

Square.h:

```
#pragma once
#include <iostream>
#include "Shape.h"

class Square :public Shape {
public:
    void Area();
};
```

Cuboid.h:

```
#pragma once
#include <iostream>
#include "Rectangle.h"

class Cuboid :public Rectangle {
private:
    double height2;
public:
    Cuboid();
    void set_height2(double);
    double get_height2();
    void Area();
};
```

Cube.h:

```
#pragma once
#include <iostream>
#include "Square.h"

class Cube :public Square {
private:
    double height1;
```



```
public:  
    Cube();  
    void set_height1(double);  
    double get_height1();  
    void Area();  
};
```

Shape.cpp:

```
#include <iostream>  
#include "Shape.h"  
  
Shape::Shape() {  
    lenth = 0;  
    width = 0;  
}  
void Shape::set_lenth(double l) {  
    lenth = l;  
}  
double Shape::get_lenth() {  
    return lenth;  
}  
void Shape::set_width(double w) {  
    width = w;  
}  
double Shape::get_width() {  
    return width;  
}
```

Rectangle.cpp:

```
#include <iostream>  
#include "Rectangle.h"  
  
void Rectangle::Area() {  
    std::cout << "Area of rectangle = " << get_lenth() * get_width() << " m^2"  
<< std::endl;  
}
```

Square.cpp:

```
#include <iostream>  
#include "Square.h"  
  
void Square::Area() {  
    std::cout << "Area of square = " << get_lenth() * get_lenth() << " m^2" <<  
std::endl;  
}
```

Cuboid.cpp:

```
#include <iostream>  
#include "Cuboid.h"  
  
Cuboid::Cuboid() {  
    height2 = 0;  
}  
void Cuboid::set_height2(double h2) {  
    height2 = h2;  
}
```



```
double Cuboid::get_height2() {
    return height2;
}
void Cuboid::Area() {
    std::cout << "Area of cuboid = " << 2 * ((get_lenth() * get_width()) +
(get_width() * get_height2()) + (get_height2() * get_lenth())) << " m^2" <<
std::endl;
}
```

Cube.cpp:

```
#include <iostream>
#include "Cube.h"

Cube::Cube() {
    height1 = 0;
}
void Cube::set_height1(double h1) {
    height1 = h1;
}
double Cube::get_height1() {
    return height1;
}
void Cube::Area() {
    std::cout << "Area of Cube = " << 6 * get_lenth() * get_lenth() << " m^2"
<< std::endl;
}
```

Main.cpp:

```
#include <iostream>
#include "Shape.h"
#include "Rectangle.h"
#include "Square.h"
#include "Cuboid.h"
#include "Cube.h"

int main() {
    Shape S;
    Rectangle R;
    Square s;
    Cuboid C;
    Cube c;
    int choice;
    while (1) {
        std::cout << "Which shape's area would you like to find?" <<
std::endl;
        std::cout << "1. Rectangle" << std::endl;
        std::cout << "2. Square" << std::endl;
        std::cout << "3. Cuboid" << std::endl;
        std::cout << "4. Cube" << std::endl;
        std::cout << "5. Exit" << std::endl;
        std::cout << "Enter your choice: ";
        do {
            std::cin >> choice;
            switch (choice) {
                case 1: {
                    double l, w;
                    std::cout << "Enter lenth: ";
                    std::cin >> l;
                    R.set_lenth(l);
                }
            }
        }
    }
}
```



National University of Sciences and Technology (NUST) School of Electrical Engineering and Computer Science

```
        std::cout << "Enter width: ";
        std::cin >> w;
        R.set_width(w);
        R.Area();
        choice = 0;
        break;
    }
    case 2: {
        double l;
        std::cout << "Enter length of one side: ";
        std::cin >> l;
        s.set_length(l);
        s.Area();
        choice = 0;
        break;
    }
    case 3: {
        double l, w, h;
        std::cout << "Enter length: ";
        std::cin >> l;
        C.set_length(l);
        std::cout << "Enter width: ";
        std::cin >> w;
        C.set_width(w);
        std::cout << "Enter height: ";
        std::cin >> h;
        C.set_height2(h);
        C.Area();
        choice = 0;
        break;
    }
    case 4: {
        double l;
        std::cout << "Enter length of one side: ";
        std::cin >> l;
        c.set_length(l);
        c.Area();
        choice = 0;
        break;
    }
    case 5: {
        choice = 0;
        std::cout << "Thank you!" << std::endl;
        return 0;
    }
    default: {
        std::cout << "Invalid Input!" << std::endl;
        std::cout << "Input again: ";
    }
}
} while (choice >= 1);

}
```



Output Screenshots

<pre>Which shape's area would you like to find? 1. Rectangle 2. Square 3. Cuboid 4. Cube 5. Exit Enter your choice: 1 Enter lenth: 5 Enter width: 6 Area of rectangle = 30 m^2 Which shape's area would you like to find? 1. Rectangle 2. Square 3. Cuboid 4. Cube 5. Exit Enter your choice: 2 Enter lenth of one side: 3 Area of square = 9 m^2 Which shape's area would you like to find? 1. Rectangle 2. Square 3. Cuboid 4. Cube 5. Exit Enter your choice: 3 Enter lenth: 12 Enter width: 3 Enter height: 5 Area of cuboid = 222 m^2</pre>	<pre>Which shape's area would you like to find? 1. Rectangle 2. Square 3. Cuboid 4. Cube 5. Exit Enter your choice: 4 Enter lenth of one side: 9 Area of Cube = 486 m^2 Which shape's area would you like to find? 1. Rectangle 2. Square 3. Cuboid 4. Cube 5. Exit Enter your choice: 5 Thank you! C:\Users\Lenovo\OneDrive\Documents\3rd Semester\Object .exe (process 46812) exited with code 0. Press any key to close this window . . .■</pre>
--	--

Task 2: How function over riding is different from function over loading?

Ans: Function overloading takes place when multiple functions in the same class or namespace have the same name but different parameters and return types. On the other hand, function overriding occurs in a hierarchy system where a derived class introduces a new implementation for a function previously defined in the base class.

Conclusion:

In this lab, we were able to understand the concepts of inheritance along with overriding functions. We were able to use overriding to our advantages and provided a unique output as our result.