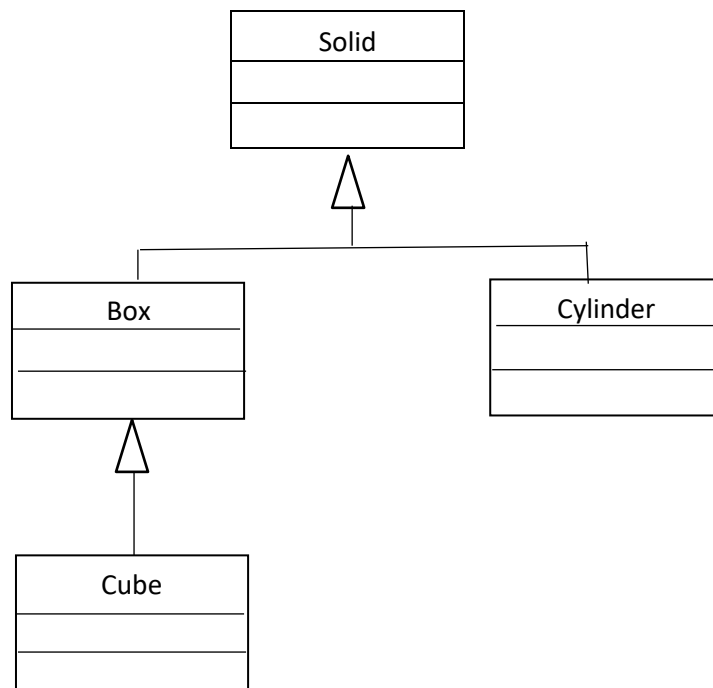


Objectives:

- Implementing Inheritance relationship.



a) Implement the Solid class

```
class Solid {
    public:
        Solid(); // default constructor
        virtual float volume();
        virtual float surfaceArea();
};
```

Note : The content of the functions and constructor is blank

- b) Implement the Box derived class.

```
class Box : public Solid
```

Hint : Use protected instead of private for the properties for all classes.

You should have the following properties.

length, width, height

Implement the *volume()* and *surfaceArea()* methods

volume = length x width x height

surfaceArea = 2x(length x width + length x height + width x height)

- c) Implement the Cylinder derived class.

```
class Cylinder : public Solid
```

Hint : Use protected instead of private for the properties for all classes.

You should have the following properties.

radius, height

Implement the *volume()* and *surfaceArea()* methods

$\text{volume} = \text{height} \times \pi \times \text{radius}^2$

$\text{surfaceArea} = \text{height} \times 2 \times \pi \times \text{radius} + 2 \times \pi \times \text{radius}^2$

- e) Implement the Cube class to derive from the Box class

```
class Cube : public Box
```

Note : You can assign to length, width and height the value mlength that you get from the constructor. You don't need to redefine `volume()` or `surfaceArea()` methods again.

```
#include "Box.h"
#ifndef _CUBE
#define _CUBE

class Cube : public Box {

public:

    Cube();
    Cube(int mlength);
};

#endif
```

Note : The `#ifndef` function is used to check if there is a definition of `_CUBE`. This is to prevent the `Cube.h` code being executed more than one time (which results in an error). Do this for the other header files as well (use unique identifiers for each header file). You do not need to redefine `volume()` and `surfaceArea()` functions in `Cube` since it is defined in the `Box` class.

Lab Exercise 12**IT1050 – Object Oriented Concepts****Semester 1, 2023**

-
- g) Implement a main program where you have objects of all the classes, calculate and print their areas and perimeters.
- h) Implement an array of Solid objects and calculate the total area and total perimeter of the shapes.

// Modify this code that prints the area and perimeter of the 3 shapes

```
Solid *solids[5];  
  
solids[0] = new Box(10, 5, 4);  
solids[1] = new Cylinder(5, 10);  
solids[2] = new Cube(20);  
  
  
for (int r=0; r<3; r++) {  
    cout << "Solids : " << r << endl;  
    cout << solids[r]->volume() << endl;  
    cout << solids[r]->surfaceArea() << endl;  
}
```