LAB REPORT (SHAPE HIERARCHY)

Group members:

- Laiba Batool
- Mariyam Muzammil
- Hira Arif

Course:	DSA
Lab Number:	01
Lab Title:	Reviewing OOP Concepts (Inheritance and Polymorphism)
Date:	29/9/24

Objective:

- Practice Objects and Classes in little more detail.
- Understand and practice the Inheritance concepts.
- Understand and implement the Polymorphism concepts.

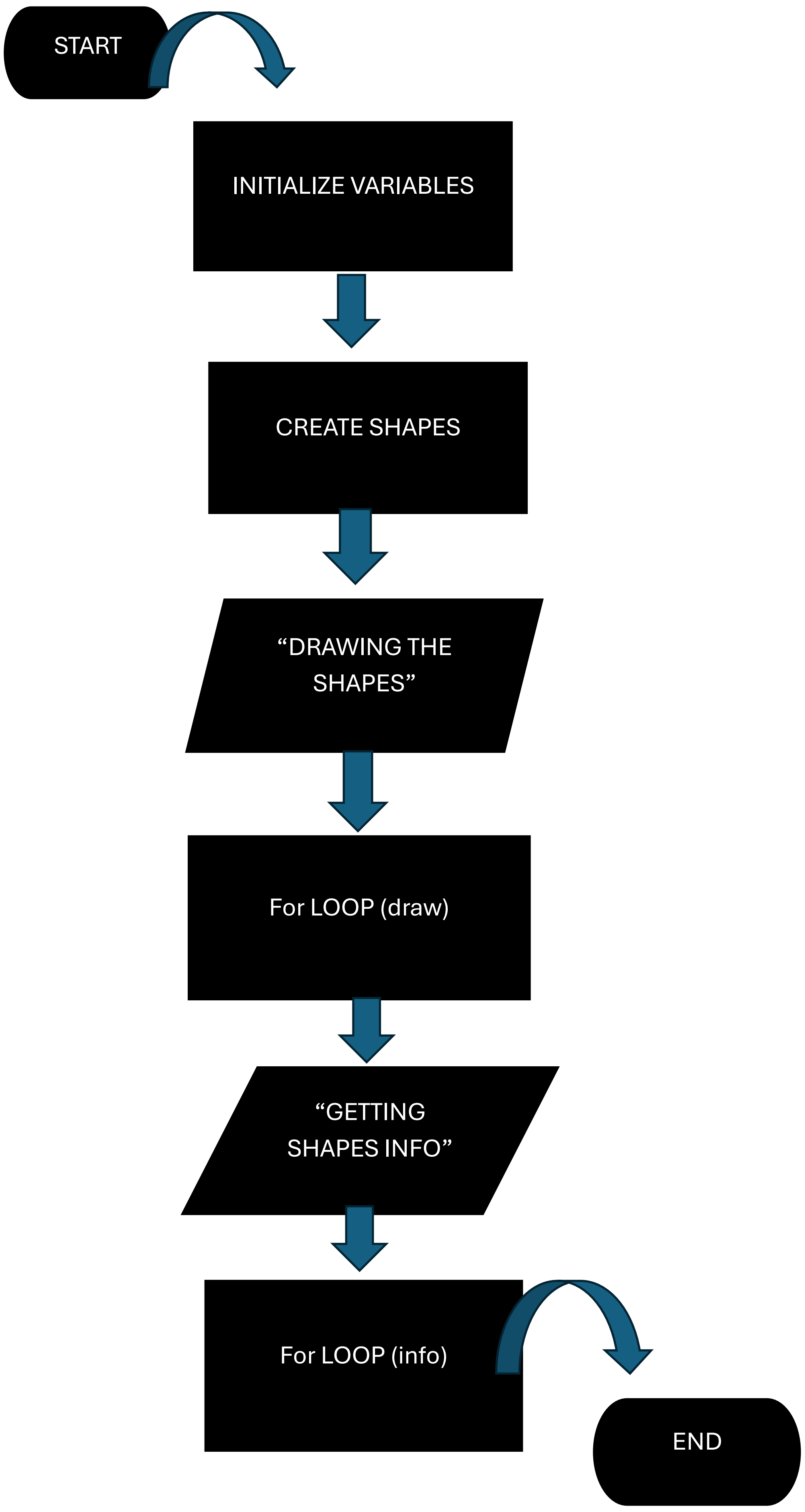
Description:

In this lab we implemented the given header files with all the functions given in UML Diagram. We also created the main function to test all the functionalities

Conclusion:

The main function is working well. All functions are good to go. We learnt a lot about inheritance and its types and abstract classes.

MAIN FUNCTION FLOWCHART:



HEADER FILES :

Shape.h

```
#ifndef SHAPE_H

#define SHAPE_H

#include<iostream>

#include<string>

using namespace std;

class Shape    //abstract class

{

   public:

      Shape(string name);   //parameterized constructor

         //member functions

   //pure virtual functions

      virtual void draw()=0;

      virtual void info()=0;

      string get_name() ;

      virtual ~Shape(); //destructor

   protected:

   private:

       string name;

};

#endif // SHAPE_H
```

Shape_2D.h

```
#ifndef SHAPE_2D_H

#define SHAPE_2D_H

#include"Shape.h"

#include<iostream>

#include<string>

using namespace std;

class Shape_2D :public Shape   //child class
```

```
{
   public:
       Shape_2D(string n_name);     //parameterized constructor
        //member functions
        virtual void info();
        virtual double calculate_area()=0; //pure virtual function
      virtual ~Shape_2D();   //destructor
   protected:
   private:
};
#endif // SHAPE_2D_H
```

Shape_3D.h

```
#include "Shape_2D.h"
#include<iostream>
#include<string>
using namespace std;
#ifndef SHAPE_3D_H
#define SHAPE_3D_H
#include"Shape.h"
#include<iostream>
#include<string>
using namespace std;
class Shape_3D :public Shape  //child class(abstract class)
{
   public:
      Shape_3D(string n_name);  //parameterized constructor
      virtual void info();      //virtual function
      virtual double calculate_volume()=0;   //pure virtual function
      virtual ~Shape_3D();
   protected:
   private:
};
#endif // SHAPE_3D_H
```

Circle.h

```
#ifndef CIRCLE_H
#define CIRCLE_H
#include"Shape_2D.h"
#include<iostream>
#include<string>
using namespace std;
class Circle :public Shape_2D
{
  public:
    //parameterized constructors
    Circle(string n_name);
    Circle(string n_name,double radius);
    //member functions
     virtual void draw();
    virtual void info();
    virtual double calculate_area();
    virtual ~Circle();   //destructor
  protected:
  private:
    double radius;
};
#endif // CIRCLE_H
```

Rectangle.h

```
#ifndef RECTANGLE_H
#define RECTANGLE_H
```

```
#include"Shape_2D.h"

#include<iostream>

#include<string>

using namespace std;

class Rectangle :public Shape_2D  //child class of shape_2D

{

   public:

       Rectangle(string name);

       Rectangle(string name,double length,double width);

       //member functions

         virtual void draw();

       virtual void info();

       virtual double calculate_area();

       virtual ~Rectangle();

   protected:

   private:

       double length;

       double width;

};

#endif // RECTANGLE_H
```

Square.h

```
#ifndef SQUARE_H

#define SQUARE_H

#include"Shape_2D.h"

#include<iostream>

#include<string>

using namespace std;

class Square :public Shape_2D
```

```
{
  public:
    Square(string name);
    Square(string name,double side);
    virtual void draw();
    virtual void info();
    virtual double calculate_area();
    virtual ~Square();
  protected:
  private:
    double side;
};
#endif // SQUARE_H
```

Cube.h

```
#ifndef CUBE_H
#define CUBE_H
#include"Shape_3D.h"
#include<iostream>
#include<string>
using namespace std;
class Cube :public Shape_3D
{
  public:
    Cube(string name);
    Cube(string name,double side);
    virtual void draw();
    virtual void info();
    virtual double calculate_volume();
```

```
        virtual ~Cube();
    protected:
    private:
        double side;
};
#endif // CUBE_H
```

Sphere.h

```
#ifndef SPHERE_H
#define SPHERE_H
#include"Shape_3D.h"
#include<iostream>
#include<string>
using namespace std;
class Sphere :public Shape_3D
{
    public:
        Sphere(string name);
        Sphere(string name,double radius);
        virtual void draw();
        virtual void info();
        virtual double calculate_volume();
        virtual ~Sphere();
    protected:
    private:
        double radius;
};
#endif // SPHERE_H
```

.CPP FILES :

Class Shape

```
#include "Shape.h"

#include<iostream>

#include<string>

using namespace std;

Shape::Shape(string n_name)  //parameterized constructor

{

  name=n_name;

}

 string Shape::get_name()   //member function to get name

{

   return name;

}

Shape::~Shape(){}       //destructor
```

Class Shape_2D

```
#include "Shape_2D.h"

#include<iostream>

#include<string>

using namespace std;


Shape_2D::Shape_2D(string n_name):Shape(n_name){}  //parameterized constructor
,constructor of Shape is also called here

void Shape_2D::info()

{

   cout<<"i am a 2D shape";
}

Shape_2D::~Shape_2D(){}   //destructor
```

Class Shape_3D

```
#include "Shape_3D.h"

#include"Shape.h"

#include"Shape_3D.h"

#include<iostream>

#include<string>

using namespace std;


Shape_3D::Shape_3D(string n_name):Shape(n_name)   //parameterized constructor

{

}

void Shape_3D::info()    //implementing member function

{

   cout<<"I am a 3D shape";

}

Shape_3D::~Shape_3D(){}   //destructor
```

Class Circle

```
#include "Circle.h"

#include"Shape_2D.h"

#include"Shape.h"

#include<iostream>

#include<string>

using namespace std;
```

```
Circle::Circle(string n_name):Shape_2D(n_name)
{


}
Circle::Circle(string name,double radius):Shape_2D(name),radius(radius)
{
}
void Circle::draw()
{
   cout<<"Drawing Circle "<<"'"<<get_name()<<"'"<<endl;
}
double Circle::calculate_area()
{
   return 3.14*radius*radius;
}
void Circle::info()
{
   cout<<"I am a Circle "<<"'"<<get_name()<<"'"<<" of radius : "<<radius<<endl;
   cout<<"I am a 2D shape"<<endl;
   cout<<"My Surface Area is : "<<this->calculate_area()<<" square units"<<endl;
}
Circle::~Circle(){}
```

Class Rectangle

```
        #include "Rectangle.h"
        #include"Shape_2D.h"
        #include<iostream>
        #include<string>
        using namespace std;
```

```
          Rectangle::Rectangle(string name):Shape_2D(name){}

          Rectangle::Rectangle(string name,double length,double
width):Shape_2D(name),length(length),width(width){}

            //implementing member functions

          void Rectangle::draw()

          {

             cout<<"Drawing Rectangle "<<"'"<<get_name()<<"'"<<endl;

          }

          double Rectangle::calculate_area()  //calculating area

          {

             return length*width;

          }

          void Rectangle::info()

          {

             cout<<"I am a Rectangle '"<<get_name()<<"' of length : "<<length<<" and width
: "<<width<<endl;

             cout<<"I am a 2D shape"<<endl;

             cout<<"My Surface Area is : "<<this->calculate_area()<<" square units"<<endl;

          }

          Rectangle::~Rectangle(){}
```

Class Square

```
#include "Square.h"

#include"Shape_2D.h"

#include<iostream>

#include<string>

using namespace std;
```

```
Square::Square(string name):Shape_2D(name){}

Square::Square(string name,double side):Shape_2D(name),side(side){}

void Square::draw()

{

   cout<<"Drawing Square "<<"'"<<get_name()<<"'"<<endl;

}

double Square::calculate_area()

{

   return side*side;

}

void Square::info()

{

   cout<<"I am a Square '"<<get_name()<<"' of side : "<<side<<endl;

   cout<<"I am a 2D shape"<<endl;

   cout<<"My Surface Area is : "<<this->calculate_area()<<" square units"<<endl;

}

Square::~Square(){}
```

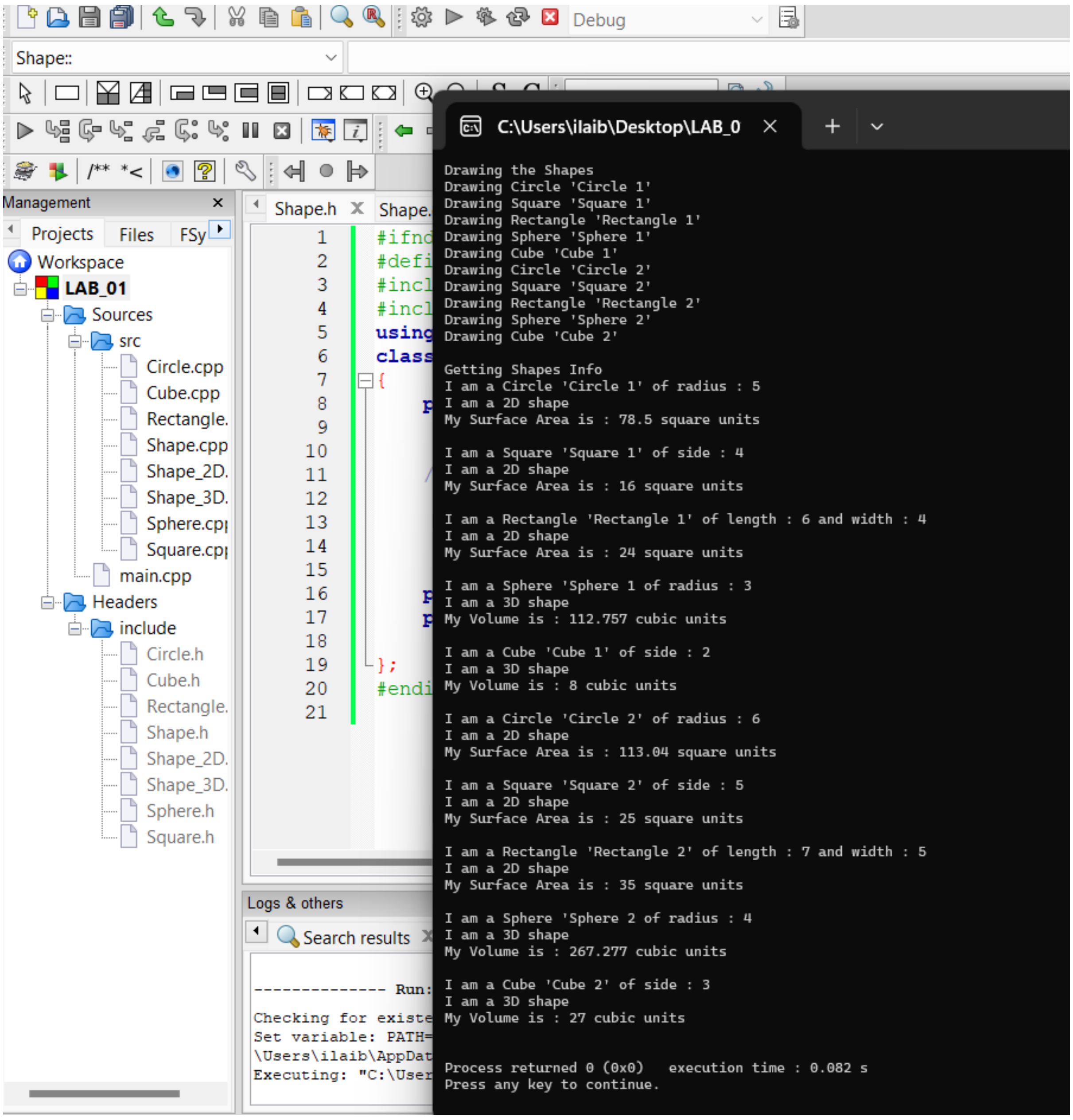
Class Cube

```
#include "Cube.h"

#include"Shape_3D.h"

#include<iostream>

#include<string>

using namespace std;

Cube::Cube(string name):Shape_3D(name){}

Cube::Cube(string name,double side):Shape_3D(name),side(side){}

void Cube::draw()

{

   cout<<"Drawing Cube "<<"'"<<get_name()<<"'"<<endl;
}
double Cube::calculate_volume()
{
   return side*side*side;
}
void Cube::info()
{
   cout<<"I am a Cube '"<<get_name()<<"' of side : "<<side<<endl;
   cout<<"I am a 3D shape"<<endl;
   cout<<"My Volume is : "<<this->calculate_volume()<<" cubic units"<<endl;
}
Cube::~Cube(){}
```

Class Sphere

```
#include "Sphere.h"
#include"Shape_3D.h"
#include<iostream>
#include<string>
using namespace std;
Sphere::Sphere(string name):Shape_3D(name){}
Sphere::Sphere(string name,double radius):Shape_3D(name),radius(radius){}
void Sphere::draw()
{
   cout<<"Drawing Sphere "<<"'"<<get_name()<<"'"<<endl;
}
double Sphere::calculate_volume()
{
   return (1.33)*3.14*radius*radius*radius;}
void Sphere::info(){
   cout<<"I am a Sphere '"<<get_name()<<" of radius : "<<radius<<endl;
   cout<<"I am a 3D shape"<<endl;
   cout<<"My Volume is : "<<this->calculate_volume()<<" cubic units"<<endl;
}
Sphere::~Sphere(){}
```

DIRECTORY STRUCTURE :



```
Workspace
  LAB_01
    Sources
      src
        Circle.cpp
        Cube.cpp
        Rectangle.
        Shape.cpp
        Shape_2D.
        Shape_3D.
        Sphere.cpj
        Square.cpj
      main.cpp
    Headers
      include
        Circle.h
        Cube.h
        Rectangle.
        Shape.h
        Shape_2D.
        Shape_3D.
        Sphere.h
        Square.h
```

```
#ifnd
#defi
#incl
#incl
using
class
{
};
#endi
```

```
-------------- Run:
Checking for existe
Set variable: PATH=
\Users\ilaib\AppDat
Executing: "C:\User
```

```
Drawing the Shapes
Drawing Circle 'Circle 1'
Drawing Square 'Square 1'
Drawing Rectangle 'Rectangle 1'
Drawing Sphere 'Sphere 1'
Drawing Cube 'Cube 1'
Drawing Circle 'Circle 2'
Drawing Square 'Square 2'
Drawing Rectangle 'Rectangle 2'
Drawing Sphere 'Sphere 2'
Drawing Cube 'Cube 2'

Getting Shapes Info
I am a Circle 'Circle 1' of radius : 5
I am a 2D shape
My Surface Area is : 78.5 square units

I am a Square 'Square 1' of side : 4
I am a 2D shape
My Surface Area is : 16 square units

I am a Rectangle 'Rectangle 1' of length : 6 and width : 4
I am a 2D shape
My Surface Area is : 24 square units

I am a Sphere 'Sphere 1 of radius : 3
I am a 3D shape
My Volume is : 112.757 cubic units

I am a Cube 'Cube 1' of side : 2
I am a 3D shape
My Volume is : 8 cubic units

I am a Circle 'Circle 2' of radius : 6
I am a 2D shape
My Surface Area is : 113.04 square units

I am a Square 'Square 2' of side : 5
I am a 2D shape
My Surface Area is : 25 square units

I am a Rectangle 'Rectangle 2' of length : 7 and width : 5
I am a 2D shape
My Surface Area is : 35 square units

I am a Sphere 'Sphere 2 of radius : 4
I am a 3D shape
My Volume is : 267.277 cubic units

I am a Cube 'Cube 2' of side : 3
I am a 3D shape
My Volume is : 27 cubic units


Process returned 0 (0x0)   execution time : 0.082 s
Press any key to continue.
```