Lab-2 Notes

Goal : Make a Shape Container which stores the shapes(Circle, Rectangle and Triangle). You can add, remove and print them.

What is given?

- You have been given input.txt and a ShapeInterface on the wiki page.

Procedure:

Now you have to think, how can you represent the shape?

You have to use Shape interface “Shape.h”, which is given. Shape interface consists of the names of the method. It does not include any constructor and definition of functions.

Shape.h contain three methods:

1. Area() - Pure Virtual
2. ShapeName() - Pure Virtual
3. Destructor

- You have to implement pure virtual functions of interface in the inherited class.

Here you have to create the three classes for three different Shapes which inherit the properties from the Shape interface.

1. Triangle (Base, Height)
2. Circle (Radius)
3. Rectangle (Length, Height)

- You have to implement the Area() and ShapeName() functions for all these three classes because they are pure virtual functions. You have to over-ride those functions according to the functionality of particular shape.

- For example, Area() function of the Triangle would be different from the Area() function of Rectangle and Circle().

Here is the example of Circle class, you can take as reference:

Circle. h

```
#ifndef CIRCLE_H
#define CIRCLE_H
#include <string>
#include "shape.h"
class circle: public shape // This is use for inheriting from Shape Interface
{
	private:
	double m_radius;

	public:

	circle(double radius);
	double area() const;
	std::string shapeName() const;
};
#endif
```

Circle.cpp

```
#include"circle.h"

circle::circle(double radius)
{
	m_radius=radius;
}

double circle::area() const
{
	return(3.1416*m_radius*m_radius);
}

std::string circle::shapeName() const
{
	return("Circle");
}
```

- Similarly, you can implement Rectangle.h, Rectangle.cpp, Triangle.h and Triangle.cpp.

ShapeContainer Class:

Now you also have been given ShapeContainer.h file.

You have to create ShapeContainer.cpp, which implement the ShapeContainer.h functions.

- ShapeContainer class is responsible for creating the array of shape pointers, adding the shape, removing the shape, returning area and shape name according the index.

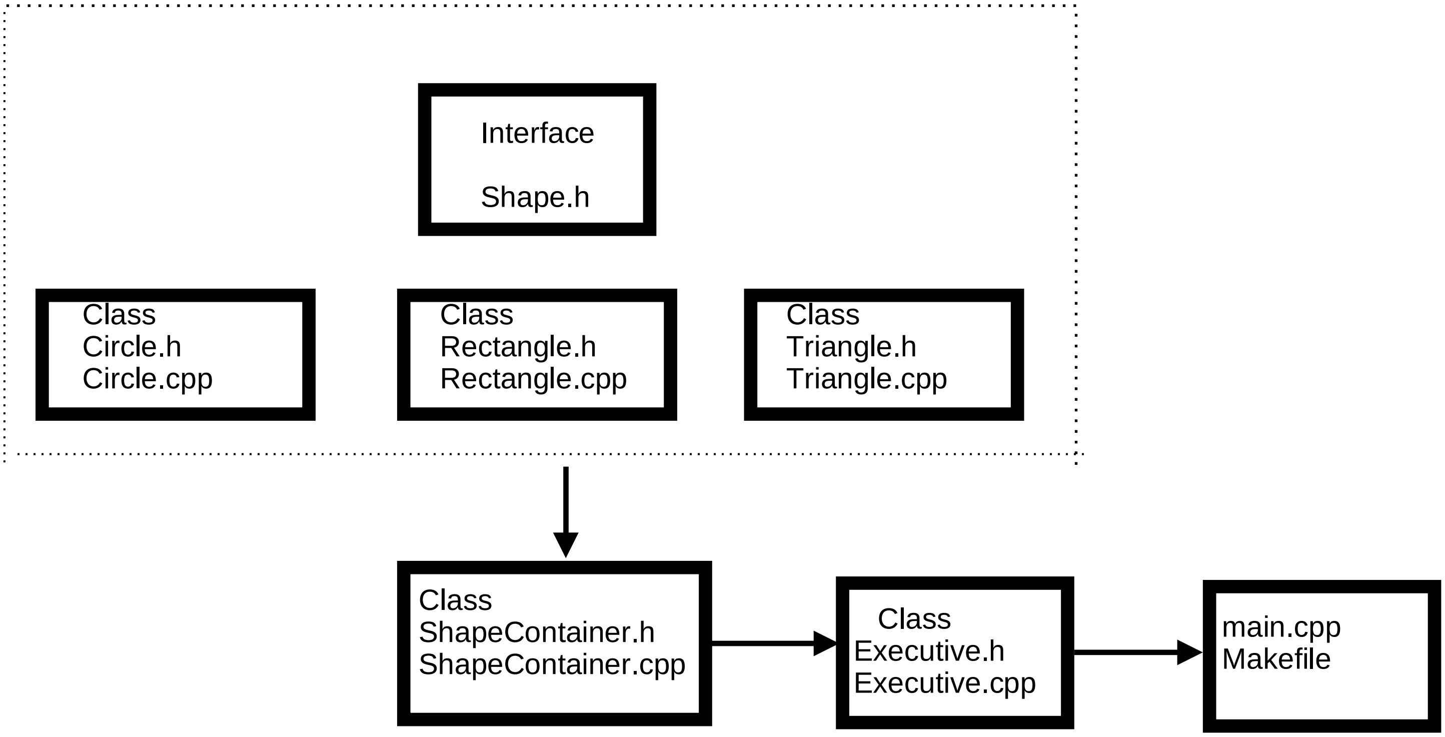
- List of the methods has been given in ShapeContainer.h in wiki page which you have to implement in ShapeContainer.cpp

Executive Class:

Executive class will be responsible for:

- reading from the input.txt file.
- it will create the ShapeContainer Object of given size.
- reading the command from the file and creating the particular shape object(Circle, Rectangle or Triangle) according to given command.

You will also require to handle the exception when the index is out of bounds or you encounter nullptr.



Here is the total number of files you would need:

main.pp
executive.h
executive.cpp
ShapeContainer.h
ShapeContainer.cpp
Shape.h
Circle.h
Circle.cpp
Triangle.h
Triangle.cpp
Rectangle.h
Rectangle.cpp
Makefile
input.txt