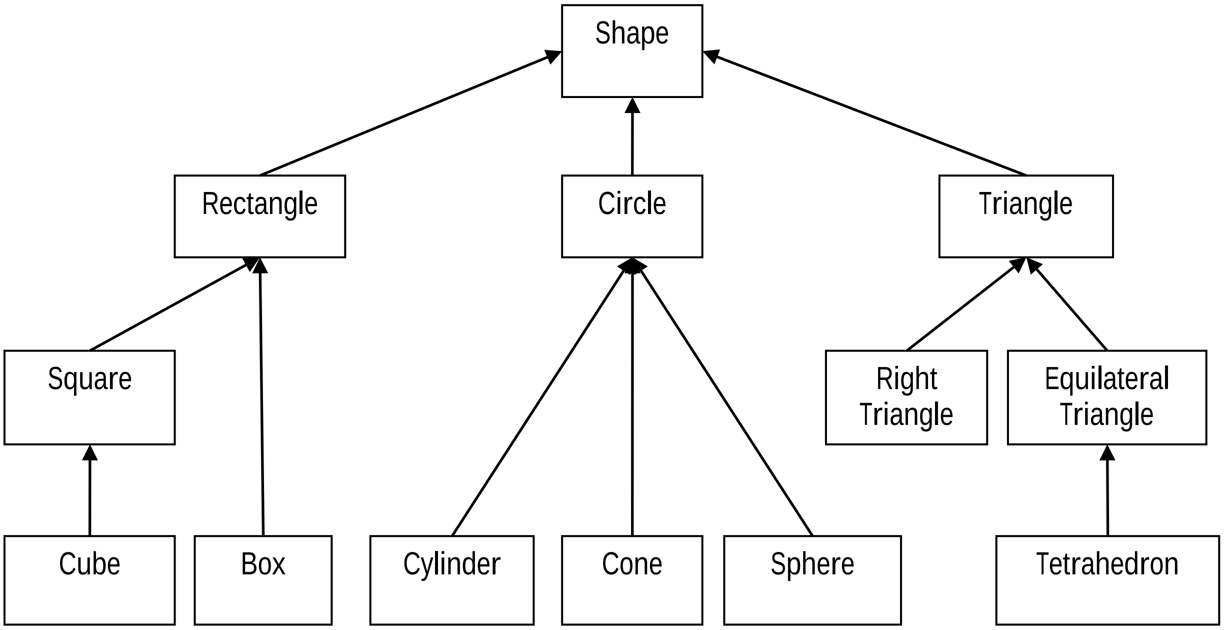
CSCI 689 **Computer Assignment 10** **Fall 2017**

(10 points)

Geometric Shapes

For this computer assignment, you are to write a C++ program to define and implement several classes for two– and three–dimensional geometric shapes. The following flowchart illustrates the hierarchical relationship between twelve different geometric shapes–six of them (rectangle, circle, triangle, square, right triangle, and equilateral triangle) are two–dimensional, and the remaining six (cube, box, cylinder, cone, sphere, and tetrahedron) are three–dimensional. Clearly more shapes can be added into this flowchart.



In this flowchart, shape indicates an abstract shape and it is represented by the abstract class Shape, which will be the base class for your design. Definition of this class is given in the header file Shape.h, in directory: ~cs689/progs/17f/p10.

For two-dimensional shapes, put the definitions of classes to represent these shapes, namely Rectangle, Circle, Triangle, Square, rightTriangle, and equTriangle, in your header file Shape2.h and the implementations of the member functions of these classes in your source file Shape2.cc. For three-dimensional shapes, put the definitions of classes to represent these shapes, namely Cube, Box, Cylinder, Cone, Sphere, and Tetrahedron, in your header file Shape3.h and the implementations of the member functions of these classes in your source file Shape3.cc.

For the first level of derived classes of two-dimensional shapes (Rectangle, Circle, and Triangle), use the following protected data members: length and width for Rectangle; radius for Circle; a, b, and c (lengths of three sides) for Triangle – all of them are type double. For each of these classes, in addition to default constructor, copy constructor, assignment operator (=), function for overloading the append operator (+=), and destructor, provide the following public member functions:

- double perimeter ( ) const: Returns the perimeter of a geometric shape.
- double area ( ) const: Returns the area of a geometric shape.
- void print ( ) const: Prints the dimensions of a geometric shape.

For the second level of derived classes for two-dimensional shapes (square, rightTriangle, and equTriangle), do not add any new data members.

For the derived classes of three-dimensional shapes, use the following private data member: height for Box, Cylinder, and Cone; and nothing for the others, and provide the following public member functions:

- double perimeter ( ) const: Returns the perimeter of a geometric shape. For the classes Box, Cube, Cylinder, Cone, and Tetrahedron, it simply returns 0, but the class Sphere does not have its own.
- double area ( ) const: Returns the total surface area of a geometric shape.
- double volume ( ) const: Returns the volume of a geometric shape.
- void print ( ) const: Prints the dimensions of a geometric shape, but the classes Sphere and Tetrahedron do not have their own.

In your source file Shape2.cc, include the header file Shape2.h by inserting the statement: #include "/home/cs689/progs/17f/p10/Shape2.h" at the top of the file, and in your source file Shape3.cc, include the header file Shape3.h by inserting the statement: #include "/home/cs689/progs/17f/p10/Shape3.h" at the top of the file.

To test your program, the source file of the driver program prog10.cc, in directory: ~cs689/progs/17f/p10, is available to you.

To compile the source files and link the generated object files with the system library routines, execute: Make N=10. For a final test of your program, execute: Make execute N=10. This will test your program and generate the output file prog10.out. The correct output file, prog10.out, is in the same directory with the source file of the driver program prog10.cc.

When your program is ready, mail its source files to your TA by executing: mail_prog.689 Shape2.cc Shape3.cc.

Use the following equations to compute perimeters, areas, and volumes of geometric shapes, where P is the perimeter, A is the area, and V is the volume of a geometric shape:

- RECTANGLE: If l is the length and w is the width of a rectangle, then $P = 2(l + w)$ and $A = lw$.

- CIRCLE: If r is the radius of a circle, then $P = 2\pi r$ and $A = \pi r^2$.

- TRIANGLE: If a, b, and c are the lengths of the three sides of a triangle, then $P = a + b + c$ and $A = \sqrt{k(k-a)(k-b)(k-c)}$ for $k = P/2$ (Heron's formula).

- SQUARE: This is a special case for a rectangle, such that $l = w$, so the class Square can be derived from the class Rectangle.

- RIGHT TRIANGLE: This is a special case for a triangle, such that $c = \sqrt{a^2 + b^2}$, so the class rightTriangle can be derived from the class Triangle.

- EQUILATERAL TRIANGLE: This is also a special case for a triangle, such that $a = b = c$, so the class equTriangle can be derived from the class Triangle.

- BOX: The class Box can be derived from the class Rectangle. If h is the height of a box, then $A = 2A_0 + hP_0$ and $V = hA_0$, where A_0 is the area and P_0 is the perimeter of the top of the box.

- CUBE: This is a special case for a box, such that $l = w = h$, so the class Cube can be derived from the class Square. If A_0 is the area of one of the faces of a cube, then $A = 6A_0$ and $V = lA_0$.

- CYLINDER: The class Cylinder can be derived from the class Circle. If h is the height of a cylinder, then $A = 2A_0 + A_1$ and $V = hA_0$, where A_0 is the area of the base and A_1 is the area of the lateral surface of the cylinder. If P_0 is the perimeter of the base of the cylinder, then $A_1 = hP_0$.

- CONE: The class Cone can be derived from the class Circle. If r is the radius of the base and h is the height of a cone, then $A = A_0 + A_1$ and $V = \frac{1}{3}hA_0$, where A_0 is the area of the base and A_1 is the area of the lateral surface of the cone. If P_0 is the perimeter of the base and s is the slant height of the cone, then $A_1 = \frac{1}{2}sP_0$ where $s = \sqrt{r^2 + h^2}$.

- SPHERE: The class Sphere can be derived from the class Circle. If r is the radius of a sphere, then $A = 4A_0$, $V = \frac{4}{3}rA_0$, and $P = P_0$, where A_0 is the area and P_0 is the perimeter of the largest cross section of the sphere.

- TETRAHEDRON: It is a pyramid with four equal faces and its each face is an equilateral triangle, so the class Tetrahedron can be derived from the class equTriangle. If a is the length of a side of a tetrahedron, then $A = 4A_0$ and $V = \frac{1}{3}hA_0$, where A_0 is the area of one of the faces of the tetrahedron and $h = \sqrt{\frac{2}{3}}a$.

Note: When you write the implementation of a member function of a derived class for a geometric shape, you must use the member functions of its base class if they are applicable. This is the basic idea of inheritance.

For your computations, define the value π as follows: #define PI (4 * atan (1)), where atan ( ) is a trigonometric function, and to use this function, you need to include the system header file <math.h> in your program files.