```cpp
#include <iostream>
#include <string.h>
#include <math.h>
#define pi 3.1416
using namespace std;
class Point {
public:
    float x, y;
    Point() {
        x = 0;
        y = 0;
    }

    Point(int x, int y){
        this->x = x;
        this->y = y;
    }

    void operator<<(char s[]){
        cout<< s << "(" << this->x << ", " << this->y << ")" << endl;
    }
};
class Shape{
protected:
    char* name;
    Point startVertex;
    float side;
public:
    Shape(){
        name = new char(1);
        strcpy(name, "");
        float side = 1;
    }
    Shape(Point vertex, float a, char *name = ""){
        this->name = new char(strlen(name) + 1);
        strcpy(this->name, name);
        startVertex = vertex;
        side = a;
    }
    ~Shape(){
        delete name;
    }
    char* getName(){
        return name;
    }
    float getAside(){
        return side;
    }
    void setAside(float a){
        side = a;
    }
    Point* getVertex(){
        return &startVertex;
    }
    void setVertex(Point pt){
        startVertex = pt;
    }
};
class TwoDimensionalShape:public Shape{
public:
    TwoDimensionalShape(): Shape(){}
    TwoDimensionalShape(Point vertex, float a, char*name = ""): Shape(vertex, a, name){}
    float findArea();
    float findPerimeter();
};

class Circle:public TwoDimensionalShape{
public:
    Circle():TwoDimensionalShape(){}
    Circle(Point ct, float r, char *ner = ""):TwoDimensionalShape(ct, r, ner){}
    float findArea(){
        return side*side*pi;
    }
    float findPerimeter(){
        return 2*side*pi;
    }
    void setCenterPoint(Point ct){
        startVertex = ct;
    }
    void setRadius(float radius){
        side = radius;
    }
    void getData(){
        startVertex<<"CP";
```

```
85          cout<<"Name: "<<name<<endl;
86          cout<<"Radius: "<<side<<endl;
87      }
88  };
89  class Square:public TwoDimensionalShape{
90  private:
91      Point B;
92      Point C;
93      Point D;
94      void calcOtherPoints(Point lt_p, float side){
95         B.x = lt_p.x + side;
96         B.y = lt_p.y;
97         C.x = lt_p.x + side;
98         C.y = lt_p.y - side;
99         D.x = lt_p.x;
100        D.y = lt_p.y - side;
101     }
102 public:
103     Square():TwoDimensionalShape(){
104         Point A(-1, 1);
105         float a = 2;
106
107         startVertex = A;
108         side = a;
109         calcOtherPoints(startVertex, side);
110     }
111     Square(Point lt_p, float a, char *name = ""):TwoDimensionalShape(lt_p, a, name){
112         setLTpoint(lt_p);
113     }
114     float findArea(){
115         return side*side;
116     }
117     float findPerimeter(){
118         return 4*side;
119     }
120     void setAside(float a){
121         side = a;
122         calcOtherPoints(startVertex, side);
123     }
124     void setLTpoint(Point lt_p){
125         startVertex = lt_p;
126         calcOtherPoints(startVertex, side);
127     }
128     void getData(){
129         startVertex<<"A";
130         B<<"B";
131         C<<"C";
132         D<<"D";
133         cout<<"Name: "<<name<<endl;
134         cout<<"A side: "<<side<<endl;
135     }
136 };
137 class RightTriangle:public TwoDimensionalShape{
138 private:
139     Point B;
140     Point C;
141     //BC талыг Ox тэнхлэгтэй parallel ги үзсэн болно.
142     void calcOtherPoints(Point tp, float a){
143         B.x = tp.x + side/2;
144         B.y = tp.y - sqrt(3)/2*side;
145         C.x = tp.x - a/2;
146         C.y = tp.y - sqrt(3)/2*side;
147     }
148 public:
149     RightTriangle(): TwoDimensionalShape(){
150         Point A(0, 1);
151         startVertex = A;
152         float a = 3;
153         side = a;
154         calcOtherPoints(startVertex, side);
155     }
156     RightTriangle(Point pt, float a, char *name = ""):TwoDimensionalShape(pt, a, name){
157         setTpoint(pt);
158     }
159     float findArea(){
160         return sqrt(3)*side*side/4;
161     }
162     float findPerimeter(){
163         return 3*side/2;
164     }
165     void setAside(float a){
166         side = a;
167         calcOtherPoints(startVertex, side);
168     }
```

```
169         void setTpoint(Point pt){
170             startVertex = pt;
171             calcOtherPoints(startVertex, side);
172         }
173         void getData(){
174             startVertex<<"A";
175             B<<"B";
176             C<<"C";
177             cout<<"Name: "<<name<<endl;
178             cout<<"A side: "<<side<<endl;
179         }
180     };
181
182     int main(){
183         Point ct(1, 1), st(-2, 2), tt(2, 2);
184
185         Circle c1(ct, 3, "C1");
186         cout<<"Circle: "<<endl;
187         c1.getData();
188         cout<<"Area: "<<c1.findArea()<<endl;
189         cout<<"Length: "<<c1.findPerimeter();
190         cout<<"\n\n";
191
192         Square s1(st, 2, "S1");
193         cout<<"Square: "<<endl;
194         s1.getData();
195         cout<<"Area: "<<s1.findArea()<<endl;
196         cout<<"Length: "<<s1.findPerimeter();
197         cout<<"\n\n";
198
199         RightTriangle rt1(tt, 2, "RT1");
200         cout<<"RightTrangle: "<<endl;
201         rt1.getData();
202         cout<<"Area: "<<rt1.findArea()<<endl;
203         cout<<"Length: "<<rt1.findPerimeter();
204         cout<<"\n\n";
205
206         return 0;
207     }
208
```