

## Exercise 4

Tasks marked with a \* are assessed coursework. Hand in your solutions to these via email to [rn@ic.ac.uk](mailto:rn@ic.ac.uk). (Resit students submit solutions to all the tasks.) Use the subject line “C++ CW: surname\_firstname\_CW4”, where `surname_firstname_CW4.cpp` is the attached file that contains your solution. The course will be assessed based on 5 pieces of coursework (25%) and an end of term driving test (75%). Your submission must be **your own work** (submissions will be checked for plagiarism), and it should compile (and run) with the GNU C++ compiler `g++`. The deadline for submitting this week’s coursework is **11/03/2016**.

1\*. *Shapes’ areas*

Write a base class `shape` for 2D geometrical shapes and derive from it at least the classes `triangle`, `circle` and `rectangle`. A circle is defined by its centre and radius, while triangles and rectangles are defined by their vertices. Here for simplicity you may assume that the sides of the rectangle are aligned with the  $x$ - and  $y$ -axis. For each of the derived classes provide methods that return the area and circumference of the shape, respectively. Your code should also work for degenerate cases, where the area is zero. In these cases the circumferences of triangle and rectangle are defined as usual, i.e. as the sum of the three (or four) sides. Your `circle` constructor should convert negative values for the radius to zero.

Parts of the base class `shape` could look something like this.

```
class shape {
protected:
    point centre;                // uses struct point { double x,y; };
public:
    shape() { centre.x = centre.y = 0.0; }
    shape(const point &p) : centre(p) {}
    virtual double area() const = 0;
    virtual double circ() const = 0;
    virtual ~shape() {}
};
```

Then the declaration of the `circle` class would include at least the following.

```
class circle : public shape {
    double radius;
public:
    circle(const point &p, double r) : shape(p), radius(r) {}
    double area() const;
    double circ() const;          // both may use constant M_PI from <cmath>
};
```

Also provide a (global) function `double sratio(const shape &s)` that computes the ratio  $r = \frac{a}{c}$  between the area  $a$  and the circumference  $c$  for a given shape. Here, for simplicity, we define  $\frac{0}{0} := 1$ . Furthermore, implement a member function `double shape::dist(const shape &s)` that computes the distance between two shapes. Here we simplify the task by defining the distance to be the distance between the centres of the two shapes. Here the centre of a triangle with vertices  $x_1, x_2, x_3$  is conveniently defined by  $\frac{1}{3}(x_1 + x_2 + x_3)$ .

Your code should be able to execute all of the following statements correctly.

```
int main() {
    point p(-15,20), p1(0,0), p2(3,0), p3(1.5,3), p4(0,1), p5(3,2);
    circle c(p,5.0);
    triangle t(p1,p2,p3);
    rectangle r(p4,p5);
    cout << sratio(r) << endl;
    cout << "The distance between triangle and circle is " << t.dist(c) << endl;
    return 0;
}
```

2. *Usage of the string class*

Banks often ask customers to write a certain amount in words. Write a program that takes an integer number between 1 and 999,999 and returns the amount in words. E.g. 43540 = “Forty three thousand five hundred and forty”.

[Hint: Visit e.g. [www.cplusplus.com/reference/string/string](http://www.cplusplus.com/reference/string/string) for details on the C++ string class.]