In this exercise we study polymorphism.

Excercise A (4p)

In this exercise we implement a class hierarchy similar to the one used as example in the lectures.

Derive two classes from Point:

1. Circle: add radius
2. Square: add width and height.

All derived classes must implement all virtual functions and define constructors that take 2 – 4 values as parameter depending on the class. All parameters must have a default value.

See example output at end of this document.

```
//definition of class Point
class Point {
public:
    //constructor
    Point(double xcoord = 0.0, double ycoord = 0.0);
    // destructor
    virtual ~Point() = default;
    virtual void input(const char* prompt); // ask values from user
    virtual void output() const; // print coordinates and area (if applicable)
    virtual double area() const; // return area. (Point returns 0.0)
    void move(double deltax, double deltay);
private:
    double x;
    double y;
};
```

Write a program that defines **a vector of shared pointers** to the base class.
Then program puts the following objects into vector:

- Point with coordinates (1.0, 1.0)
- Circle at (2.0, 2.0) with radius 2.0
- Square at (5.0, 5.0) with width and height of 2.0
- Square whose data is asked from user before adding to the vector
- Circle whose data is asked from user before adding to the vector
- Point whose data is asked from user before adding to the vector

When vector has been filled program does the following:

1. Prints the object data
2. Sorts the vector by area
3. Prints the object data

See example output on the following page.

Example output:

```
Coordinates: (1,1)
Circle's area: 12.5664 Coordinates: (2,2)
Square's area: 4 Coordinates: (5,5)
Sorted:
Coordinates: (1,1)
Square's area: 4 Coordinates: (5,5)
Circle's area: 12.5664 Coordinates: (2,2)
```

Example output of calls to input():

```
Point:
Enter x:2.3
Enter y:5.7
Circle:
Enter x:3.4
Enter y:4.5
Enter radius: 6.7
Square:
Enter x:1.2
Enter y:2.3
Enter height: 3.4
Enter width: 4.5
```

Exercise B (Extra 2 p)

Make program add five random objects to the vector. The object data must be asked from user before adding the object to the vector.