CAD Exchanger

C++ assignment

Design a small program in C++ that would implement the following:

1. Support a few types of 3D geometric curves – circles, ellipses and 3D helixes. (Simplified definitions are below). Each curve should be able to return a 3D point and a first derivative (3D vector) per parameter *t* along the curve.
2. Populate a container (e.g. vector or list) of objects of these types created in random manner with random parameters. Ensure that the container will contain objects of all supported types.
3. Print coordinates of points and derivatives of all curves in the container at t=PI/4.
4. Populate a second container that would contain only circles from the first container. Make sure the second container shares (i.e. not clones) circles of the first one, e.g. via pointers.
5. Sort the second container in the ascending order of circles' radii. That is, the first element has the smallest radius, the last - the greatest.
6. Compute the total sum of radii of all curves in the second container.

Additional optional requirements:

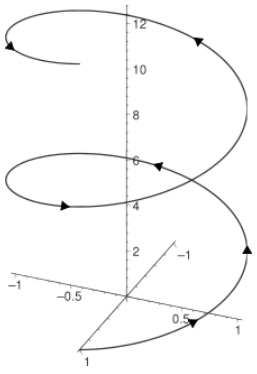
7. Split implementation into a library of curves and executable which uses API of this library.

Expectations to the implementation:

1. The implementation must use virtual methods.
2. Has neither explicit memory deallocation nor memory leaks.
3. Curves must be physically correct (e.g. radii must be positive).
4. Containers and sorting must be implemented using STL (C++ Standard Template Library).
5. The implementation may use constructs of C++11 or higher.
6. The code must compile with any compiler of your choice (gcc, Visual C++, etc).
7. The result should be sent via email or posted on github. The delivery must contain source code only (no compiled binaries): *.h and *.cpp files and project files, so that we can compile the result by ourselves.

Curve definitions:

- All curves are parametrically defined, i.e. a point is calculated using some C(t) formula.
- Circle is planar in the plane XoY (i.e. all Z-coordinates are 0) and is defined by its radius.
- Ellipse is planar in the plane XoY and is defined by its two radii, along X and Y axes.
- Helix is spatial and is defined by its radius and step (see the figure below). It takes 2 * PI in parametric space to make a round, i.e. any point on helix satisfies the condition C(t + 2*PI) = C(t) + {0, 0, step}.


12
10
8
6
4
2
−1
−1
−0.5
0.5
1
1