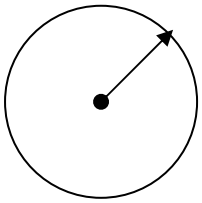
07.03.2005

Object Oriented Programming 1st Assignment

In this assignment you are asked to find the corner coordinates of the triangle formed by three tangent lines of a circle. You are obliged to model the problem in an object oriented manner. The basic steps for solving the problem are described below. Your code should include equivalent solutions for these steps.

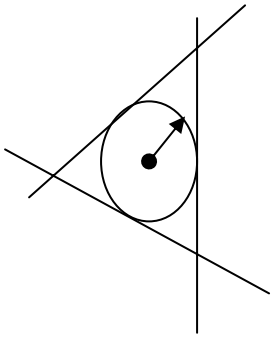
In order to find the tangent triangle of a circle, we need:

1) A circle



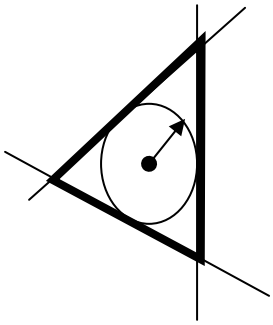
The circle will be represented by a class.
It should have two basic attributes. A center point (which will have x and y coordinates) and a radius length.

2) Three tangent lines



Tangent line will be also represented as a class. In geometry, a line can be represented in many ways. But in our case, each line should be represented by a point in space (which is the tangent point on your circle) and its slope (which is perpendicular to the normal to that point).

3) A triangle



At the last step, a triangle is formed by your three tangent lines. Here the triangle is another class that **has** three lines inside. This means that the triangle class and line class will have a "**has a**" relationship between them.
As a result, your program will calculate and print the coordinates of the corners of this triangle.

Here are the class declarations you should use:

```
class Line
{
    public:
        Line(double, double, double); //Non-vertical
        Line(double=0,double=0);      //Vertical line
        Line(const Line &);           //Copy Constructor
```

```
            double getX();
            double getY();
            double getM();

            bool onLine(double,double); //Point is on line?
 };

class Circle
{
     public:
            Circle(double, double, double);
            bool onCircle(double,double); //Point is on circle?
            Line getTangent(double x,double y);
};

class Triangle
{
     public:
         Triangle(const Line &,const Line &,const Line &);
         void getCorners();
};
```

Hints:

- The equation of a circle is $(x-h)^2 + (y-k)^2 = r^2$ where **(h,k) is the center** and **r is the radius** of the circle.

- The equation of a line can be derived from **m = (y - Y)/(x -X)** where **X and Y are coordinates** of a point on the line and **m is the slope** of the line.

- To find the intersection point of lineA and lineB you can calculate the x coordinate by
  $x= ((m_B*x_B) - (m_A*x_A) + y_A - y_B) / (m_B - m_A)$
  and y coordinate by substituting x
  $y= m_A*(x-x_A)+y_A$

- Comparison of two floating points is a tough issue and things can get complicated. Better use an offset value for comparison. That is, for example you may have a piece of code in your assignment that tests if a point is on your circle:
  Instead of:
  `if( (pow((px-centrx),2) + pow(py-centry,2) == pow(radius,2)) )`
  you should better check if
  `if( fabs(pow((px-centrx),2)+pow(py-centry,2)-pow(radius,2)) < 0.001 )`
  holds.

- Do not forget that that the **test inputs should be error free**. That means there will be no parallel lines as tangent inputs.

- **Your code should be able to run without getting an input from a user**. That is, you should define the points on the circle in your code and print them with the result of the program. However, after this initial run with default parameters, you should also ask the user if he/she wishes to continue with his own test inputs.

- **Think of a way to handle vertical tangents** (because their slope is infinite). Pay attention to the declaration of line class' constructors.

SUBMISSION:

- **Pay attention to using object oriented concepts**. The main evaluation criteria will be your success on using them. Read C++ Programming Standards, which can be found on the course web page. Write your program according to these rules. Don't forget to include sufficient explanation into your code, most essentially the compiler name at the first lines.

- **Though it is good programming practice to use header files, please keep all your source code in one file**.

- Use your student number to name that **single** file. For example, 040010601.cpp.

- Copy that **single source file** via anonymous ftp to **akasya.cs.itu.edu.tr** in oop directory. Be sure to get messages like "Transfer completed" or "xxx bytes transferred successfully" after sending your homework. Submissions will be between **9:00 and 17:30 on 21/03/2005**.

- Late submissions are not accepted.

- List files and check the size of your file. Remember that you are responsible for the correct delivery of your homework, no excuses due to transfers will be accepted after due date.

- You cannot view, delete, modify, overwrite or get files from this directory. (Even your homework!)

- For any further assistance consult Tolga Ovatman or Melike Erol in office hours.

Last but not least,

- **Cheating is strictly forbidden**. If cheating is discovered, all responsible students will be punished with a score of -10 points. It is allowed to discuss the solution of the problem with your classmates. However, assignments are not group works, they should represent your own **individual effort**.