

The Point Class

```
classdef Point < handle
% Operations on points in the plane
    properties
        % A point has an x and y coordinate
        x % the x-coordinate
        y % the y-coordinate
    end

    methods
        % Construct a point object...
        function P = Point(x,y)
            if nargin==2
                P.x = x;
                P.y = y;
            end
        end

        % Rotation...
        function Q = Rotate(ThisPoint,phi)
            % Q is a point obtained by rotating ThisPoint by phi degrees
            % counterclockwise.
            % A call of the form P.Rotate(phi) rotates point P by phi degrees
            % counterclockwise
            theta = phi*pi/180;
            c = cos(theta);
            s = sin(theta);
            x = ThisPoint.x;
            y = ThisPoint.y;
            if nargin==1
                Q = Point(c*x - s*y, s*x + c*y);
            else
                ThisPoint.x = c*x - s*y;
                ThisPoint.y = s*x + c*y;
            end
        end

        % Translation...
        function Q = Translate(ThisPoint,delX,delY)
            % Translates ThisPoint by amount delX is the x-direction and delY in
            % the y-direction.
            % A call of the form P.Translate(delX,delY) translates point P by
            % amount delX is the x-direction and delY in the y-direction.
            x = ThisPoint.x;
            y = ThisPoint.y;
            if nargin==1
                Q = Point(x+delX,y+delY);
            else
                ThisPoint.x = x+delX;
                ThisPoint.y = y+delY;
            end
        end
    end
end
```

```

% Distance
function d = Dist(ThisPoint,Q)
% The distance from ThisPoint to object Q.
% Q must be either a Point or a Circle
    if strcmp(class(Q), 'Point')
        d = sqrt((ThisPoint.x-Q.x)^2 + (ThisPoint.y-Q.y)^2);
    elseif strcmp(class(Q), 'Circle')
        d = abs(ThisPoint.Dist(Q.center) - Q.r);
    end
end

% Nearest
function P = Nearest(Q,P1,P2)
% Q, P1, and P2 are points.
% P is on the line segment that connects P1 to P2 and is the nearest
% such point to Q.
    t = ((Q.x-P1.x)*(P2.x-P1.x)+(Q.y-P1.y)*(P2.y-P1.y))/...
        ((P2.x-P1.x)^2 + (P2.y-P1.y)^2);
    if t<0
        P = P1;
    elseif t>1
        P = P2;
    else
        P = Point(P1.x+t*(P2.x-P1.x),P1.y+t*(P2.y-P1.y));
    end
end

% Separation...
function alfa = DiffSide(ThisPoint,ThatPoint,P1,P2)
% P1 and P2 are distinct points.
% alfa is true if and only if the line through P1 and P2 separates
% ThisPoint and ThatPoint.
    p1 = [P1.x P1.y];
    p2 = [P2.x P2.y];
    q = [ThisPoint.x ThisPoint.y];
    r = [ThatPoint.x ThatPoint.y];
    f = ((p1(1)-q(1))*(p1(2)-p2(2)) - (p1(2)-q(2))*(p1(1)-p2(1)))/...
        ((r(1)-q(1))*(p1(2)-p2(2)) - (r(2)-q(2))*(p1(1)-p2(1)));
    alfa = 0<=f && f<=1;
end

% Display
function Show(ThisPoint,color)
% Displays the point ThisPoint.
% The color is specified by the character c.
% Assumes hold is on.
    plot(ThisPoint.x,ThisPoint.y,['.' color], 'Markersize',30)
end

end %methods
end %classdef

```

The Circle Class

```
classdef Circle < handle
% Operations on circles
properties
% A Circle object has a center and a radius.
% The center is a Point object.
    center = Point();
    r
end

methods
% Construct a Circle object...
function C = Circle(r,P)
% Creates a circle object with radius r and center P
    C.r = r;
    C.center = P;
end

% Translation...
function A = Translate(ThisCircle,delX,delY)
% A is the circle obtained by translating the center of ThisCircle by
% an amount delX in the x-coordinate and delY in the y-coordinate.
% A call of the form B.Translate(delX,delY) translates the center
% of Circle B by an amount delX in the x-coordinate and
% delY in the y-coordinate.
    if nargin==1
        C = ThisCircle.center.Translate(delX,delY);
        A = Circle(ThisCircle.r,C);
    else
        ThisCircle.center.Translate(delX,delY);
    end
end

% Distance...
function d = Dist(ThisCircle,P)
% The distance from ThisCircle to Point P
    d = abs(P.Dist(ThisCircle.center) - ThisCircle.r);
end

% Display...
function Show(ThisCircle,c)
% Displays the circle ThisCircle.
% The color is specified by the character c.
% Assumes hold is on.
    theta = linspace(0,2*pi);
    xc = ThisCircle.center.x;
    yc = ThisCircle.center.y;
    r = ThisCircle.r;
    plot(xc+r*cos(theta),yc+r*sin(theta),c)
end
end %methods
end %classdef
```

Example Using the Point and Circle Class

```
% ShowPointCircle
% Manipulating Point and Circle Objects
clc
clear
close all
figure
plot([-5 5],[0 0],':k',[0 0],[0 0],[-5 5],':k')
axis equal square
hold on

% Create and display a point...
P = Point(3,0);
P.Show('r')

% Rotate it and display it...
P.Rotate(45);
P.Show('r')

% Get a new point through translation...
P1 = P.Translate(0,-5);
P1.Show('g')

% Get another new point through translation...
P2 = P.Translate(-5,-4);
P2.Show('g')

% Find the nearest point on a line segment
plot([P1.x,P2.x],[P1.y P2.y], 'g')
R = Nearest(P,P1,P2);
R.Show('b')

% Create a circle...
C1 = Circle(1,Point(0,0));
C1.Show('m')

% Create another circle through translation...
C2 = C1.Translate(R.x,R.y);
C2.Show('m')

% Compute some distances...
PtoR = P.Dist(R);
PtoC2 = P.Dist(C2);
title(sprintf('PtoR = %5.3f    PtoC2 = %5.3f',PtoR,PtoC2),'fontsize',18)
```