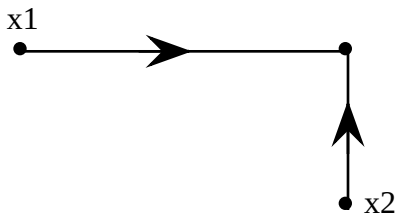
# ECE 257 - LESSON 9
# DOCUMENTATION AND TESTING OF PROGRAMS

**SPRING 2007** **A.P. FELZER**

## IN CLASS

### PROBLEM:

Given two toy cars on a collision path as follows




with the following information inputted by the user

(1) x1 and x2 equal to the starting poisitions of the two toy cars in feet
(2) v1 and v2 equal to the constant speeds of the two toy cars in feet/sec

Write a program to

(1) Plot the distance between the cars as a function of time
(2) Find the minimum distance between the cars

### STEPS IN WRITING THE PROGRAM

**STEP 1:** Write a comment describing the problem, specify the input and output variables of the program and write comments describing the steps in the algorithm

```
% Script name:  car_distance
% This program finds and plots the distance between two toy cars on a collision path and finds the
% minimum distance between them

%   Record of Revisions
%   Name                    Date            Description
%

%   Input variables
%   x_start = column vector of starting positions of the toy cars in feet
%   v = column vector of speeds of the toy cars in feet/sec
%   plot_time = time (in sec) the distance between the toy cars is to be plotted

%   Output variables
%   car_dist = distance between the toy cars as a function of time
%   min_car_dist = minimum distance between the cars

%   Internal variables
%

%   Input the speeds and starting positions
```

```
%
%   Tme variable in seconds
%
%   Toy car positions as a function of time in seconds
%
%   Distance between the toy cars as a function of time in seconds
%
%   Minimum distance between the cars in feet
%
%   Graph of the distance between the toy cars as a function of time in seconds
%
```

**STEP 2:** Come up with tests that you know or can independently find the answers to

TEST 1: Same starting distances and same speeds: both toy cars reach the origin at the same time

TEST 2: Different speeds but both cars reach the origin at the same time: x1=10, v1=4, x2=5, v2=2

TEST 3: Faster car reaches the origin first: x1=5, v1=4, x2=5, v2=2

TEST 4: Slower car reaches the origin first: x1=4, v1=2, x2=10, v2=4

**STEP 3:** Code the algorithm

```
% Script name:  car_distance
% This program finds and plots the distance between two toy cars on a collision path and finds the
% minimum distance between them

%   Record of Revisions
%   Name                    Date                Description
%
%

%   Input invariables
%   x_start = column vector of starting positions of the toy cars in feet
%   v = column vector of speeds of the toy cars in feet/sec
%   plot_time = time (in sec) the distance between the toy cars is to be plotted

%   Output variables
%   car_dist = distance between the toy cars as a function of time
%   min_car_dist = minimum distance between the cars

%   Internal variables
%   t = time in seconds
%   x = column vector of distance of the toy cars from the origin as a function of time

%   Input starting positions and speeds
x_start = input('Enter a column vector of the starting positions of the toy cars in feet:  ');
v = input('Enter a column vector of the speeds of the toy cars in feet/sec:  ');
plot_time = input('Enter the time in seconds you want to plot the distance between the toy cars:  ');
```

```
%   Time variable in seconds
t = 0: 0.01: plot_time;

%   Positions of the toy cars as a function of time in seconds
x = x_start - v*t;

%   Distance between the toy cars as a function of time
car_dist =

%   Minimum distance between the cars in feet and the time at which it occurs in seconds after start


%   Graph of the distance between the two toy cars
plot(   )
xlabel(    )
ylabel(     )
title(    )
```

**STEP 4:** Test the program - see if you get the same answers that you independently calculated in Step 2