CMPS2143 Object Oriented Programming **Assignment 2: 100 points**
DUE: Main *algorithm in the form of comments and/or code* in a .cpp file: Friday, 9/22/2017
Final program: Monday, 10/2/ 2017,

Purpose: To design classes; to use multiple classes, specification files and classes that use other classes; to use objects, arrays, and arrays of objects in a program; to read from a data file; to write to an output file and to format real output.

Method: In homework 2, you had to decide on your classes. Finish/correct the design of the classes! THEN code the implementation of your classes and write the main program.

Problem: SpaceAsteroid, Inc. wants a software application for its mining spaceships, which collect asteroid**s**. (We will consider a two-dimensional space environment that has many asteroids and one spaceship.) The spaceship has a position (x and y coordinates) and is facing a direction (Port, Bow, Starboard, Stern). The spaceship can change its position, as well as change its direction. A spaceship will search a "field" of asteroids and go to the closest asteroid and collect it – IF THE ASTEROID IS SMALL ENOUGH (that is, less than a certain SIZE.) The spaceship needs to be able to determine how far the spaceship is from each asteroid. It uses this information to decide the order in which asteroids are collected. In our *simple* version of the program, we will not worry about real spaceship movement. Your problem is only in helping the spaceship control make decisions. Then the distances between the spaceship's new position and all the remaining asteroids must be computed again, and the spaceship will go the next closest asteroid. It will continue to do this for the number of asteroids that the user wants collected. (For example, there may be 100 asteroids in the file, but the user may only want to collect 25.)

The spaceship also knows how many asteroids it has currently collected (initially it has collected 0). Of course, the number of asteroids it has collected will vary. The spaceship should at all times be able to signal back to earth its position, direction, and the number of asteroids that it has, how far it travelled to each asteroid, and where each asteroid was originally located. ***You will need to declare and use an array of asteroids.***

The original position of each asteroid is described by x and y coordinates on a 100x100 grid. (The points on a grid are 10 miles apart – so we are talking about 100,000 square miles.) An asteroid will also "know" whether or not it has been collected by the spaceship as yet. (In a real application, the environment would be 3D and the spaceship would have range sensors to locate asteroids, but we will handle the problem by making each asteroid "provide" its location.) An asteroid also "knows" how large it is.

Input: The user should either be able to use the initial default position of the spaceship or input the initial position of the spaceship at the keyboard. You will need to read a file that contains the locations and sizes for many asteroids. (Default positions will NOT be ***used*** for asteroids in this program, BUT you should still have a default constructor!) The user should input the number of asteroids it wants the spaceship to collect (up to the number of asteroids in the file) at the keyboard. Once these values are input, there are no other inputs. Turn on the spaceship and let the spaceship go!

Use a while not end of file loop to read the input from a file that will be posted on D2L. The input file looks similar to the following:

```
4   5   3                                   (x, y, size)
65  70  5.6
20  30  6.4
21  47  2.3
:
:
```

Output: The output should consist of the number of asteroids collected, the order in which the asteroids were collected, the positions of each asteroid collected, the size of the asteroid, the distance the spaceship traveled between each asteroid, and the total distance covered, as well as the total size of asteroids. I also *like* interactive introductory and exit messages to the user as they run the program. *All output should be labeled!!! For example:*

```
20 asteroids collected

Asteroid collected    position  size distance (in miles)
   -                  (0, 0)                      0.00
   2                  (10, 15)     3.4          180.30
   1                  (15, 15)     5.3          50.00
   4                  (50, 75)     6.2          694.60
   :                     :

Total distance covered:  4924.90 miles
```

Turn in:

1. Printout of *all* program files (class .h and .cpp files, main program)
2. Two outputs with spaceship starting in default position collecting 20 asteroids and a user-chosen position collection some other number of asteroids. BUT MAKE SURE IT RUNS ON OTHER INPUTS!
3. Flash drive with project files on it.
4. A class diagram.

The final program will be graded based on understandability of prompts, clarity of output, comments, indentation style, use of concepts in the purpose statement and, of course, correctness.

Extra credit – if the rock is too big, blast it and break it into 2-3 pieces and have them "move" to some other part of space. You can use random number generator to decide how many pieces to break them up into, what sizes the pieces are (total should equal the size of original asteroid), and where to "move" them to. If the new asteroid pieces should be randomly moved to the spaceship position – the spaceship will be damaged and have to halt before collecting all specimens. (5 points) I do not give help on extra credit!