**Software brief**

Write a C++ program that I can compile in Visual Studio 2012.

SW interrogates COM6, 4800 baud, 8 databits, no parity, 1 stop bit, no handshake, no error checking.

Start Main function:

1. Open the port
2. Define a variable called 'Shutdown' and initialise to 0.
3. Define an outputstream to a file called c:\autonomo\web\tmp\WeatherStation.m
   Note: this file will constantly be overwritten in the following Loop

Then the program goes through a <u>continuous loop</u> as followed:

1. Read the serial data stream; see Notes below for an example of the data stream
   Example of another program parsing the stream (not in C++ obviously):
   SerialInRecord (ComRS232,WeatherRcvdStr,36,400,0,NBytesReturnedWeather,01)
   // 36 is the $ character and this is the BeginWord
   // 400 is NBytes and means the number of bytes that should be stored in Dest after the BeginWord has been received
   // 0 is the EndWord, and means here the data stored in Dest will be NBytes after the BeginWord
   // 01 is the SerialInRecOption and means most recent record in serial buffer, store NAN if no record
2. //Checking whether string is NaN
   If WeatherRcvdStr <>"NAN" Then
   SplitStr (WeatherDataStr(),WeatherRcvdStr,CHR(36),7,5)
   //CHR(36) is the $ sign and is the split string
   //7 is NumSplit and is the maximum number of strings or values returned by the
   // instruction
   //5 is the SplitOption parameter and is a code used to specify the method of splitting
   // the string. Here it means FOOTERFILTER - Any string preceding FilterString is
   // returned in SplitResult

   Else
   wait for 200 milliseconds and restart the loop (don't do any of the following code)

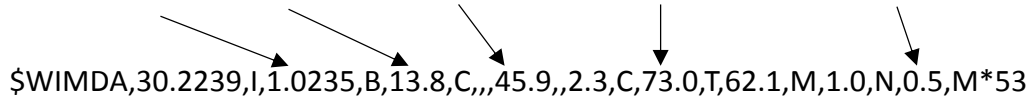
3. // now do GPGSA NMEA decoding
   SplitStr (NMEACodeStr(),WeatherDataStr(1),CHR(42),2,7)
   //CHR(42) is the * sign and is the split string
   //2 is NumSplit and is the maximum number of strings or values returned by the instruction
   //7 is the SplitOption parameter and is a code used to specify the method of splitting the string. Here it means FOOTERFILTERCHARS - Strings preceding any character in the FilterString char list are returned in SplitResult. Redundant delimiters are treated as a single

delimiter. In addition, multiple delimiters can be specified; e.g., ":;" will split a string based on the colon or semicolon

4. See Notes for an explanation of the NMEA messages; we need to write the following semi-colon separated parameters into the WeatherStation.m file:
   BarPressBar;AirTempC;RelHumid;WindDegTrue;WindSpeedMetre;timestamp<cr>

   Here is where the values for each parameter can be found in the WIMDA message:

   BarPressBar AirTempC RelHumid WindDegTrue WindSpeedMetre

   $WIMDA,30.2239,I,1.0235,B,13.8,C,,,45.9,,2.3,C,73.0,T,62.1,M,1.0,N,0.5,M*53

   timestamp should be coded as followed:

```
__int64 Counter;
__int64 Frequency;
double ThisTimeStamp;
QueryPerformanceFrequency((LARGE_INTEGER *) &Frequency);
QueryPerformanceCounter((LARGE_INTEGER*)&Counter);
ThisTimeStamp = (double)(Counter)/(double)Frequency*1000;
```

   Notes:
   - above variables are being overwritten every time the loop is executed.

5. Sleep(500)

Loop is terminated once 'Shutdown' is set to 1 – you do not have to worry how it gets set to 1.

Close Port and any other cleanup tasks required.

**Notes:**

1. Code needs to check for any errors, eg. opening / closing the port, etc..
2. Sample stream:

```
2016-08-27 10:20
        $GPRMC,002050.00,A,3645.8337,S,14419.2885,E,0.0,222.5,270816,10.9,E,A*1F
2016-08-27 10:20
        $WIMDA,30.2269,I,1.0236,B,13.8,C,,,45.9,,2.3,C,80.6,T,69.7,M,1.2,N,0.6,M*53
2016-08-27 10:20        $HCHDG,293.5,0.0,E,10.9,E*77
2016-08-27 10:20        $WIMWV,136.9,R,1.3,N,A*2C
2016-08-27 10:20        $WIMWD,81.1,T,70.2,M,1.2,N,0.6,M*52
2016-08-27 10:20        $GPGSA,A,3,22,23,1,14,31,32,11,3,26,,,,2.0,1.1,1.6*36
2016-08-27 10:20
        $GPRMC,002051.00,A,3645.8337,S,14419.2887,E,0.0,16.5,270816,10.9,E,A*29
2016-08-27 10:20
        $WIMDA,30.2239,I,1.0235,B,13.8,C,,,45.9,,2.3,C,73.0,T,62.1,M,1.0,N,0.5,M*53
2016-08-27 10:20        $HCHDG,293.5,0.0,E,10.9,E*77
2016-08-27 10:20        $WIMWV,144.5,R,1.0,N,A*26
2016-08-27 10:20        $WIMWD,91.0,T,80.1,M,1.0,N,0.5,M*5F
```

2016-08-27 10:20 $GPGSA,A,3,22,23,1,14,31,32,11,3,26,,,,2.0,1.1,1.6*36
2016-08-27 10:20
$GPRMC,002052.00,A,3645.8337,S,14419.2888,E,0.0,201.3,270816,10.9,E,A*17
2016-08-27 10:20
$WIMDA,30.2239,I,1.0235,B,13.8,C,,,45.9,,2.3,C,82.0,T,71.1,M,0.7,N,0.4,M*58
2016-08-27 10:20 $HCHDG,293.5,0.0,E,10.9,E*77
2016-08-27 10:20 $WIMWV,133.0,R,0.8,N,A*2A
2016-08-27 10:20 $WIMWD,75.9,T,65.0,M,0.8,N,0.4,M*5E
2016-08-27 10:20 $GPGSA,A,3,22,23,1,14,31,32,11,3,26,,,,2.0,1.1,1.6*36
2016-08-27 10:20
$GPRMC,002053.00,A,3645.8337,S,14419.2888,E,0.0,14.2,270816,10.9,E,A*21
2016-08-27 10:20
$WIMDA,30.2239,I,1.0235,B,13.8,C,,,46.0,,2.3,C,68.0,T,57.1,M,0.9,N,0.5,M*5D
2016-08-27 10:20 $HCHDG,293.6,0.0,E,10.9,E*74
2016-08-27 10:20 $WIMWV,121.8,R,0.9,N,A*20
2016-08-27 10:20 $WIMWD,66.6,T,55.7,M,0.8,N,0.4,M*57
2016-08-27 10:20 $GPGSA,A,3,22,23,1,14,31,32,11,3,26,,,,2.0,1.1,1.6*36
2016-08-27 10:20
$GPRMC,002054.00,A,3645.8337,S,14419.2889,E,0.0,127.1,270816,10.9,E,A*15