1a.)

(a+jb)(c+jd)= ac+ajd+jbc-bd=(ac-bd)+j(ad+bd)

k1= c(atb)

k2= a(a-c)

k3= b(c+d)

ac-bd= k1-k3

bc+ad= k1+k2

1b.) O(N^2) is important because it estimates how fast the overall execution time grows as the size of the input grows. This is important In a Discrete Fourier transform because it has a growth rate of O(n^2). As the number of input samples grows the difference in computing the DFT can be noticed. As N grows the computational speed is faster. W^N is periodic with a period N and is symmetric about the imaginary axis. The symmetry allows the number of computations for a DFT to be reduced significantly. By splitting the DIT in two sequences the N point can be obtained by two N/2 DFTs. The divide and conquer theory also works in that the N point DFT is successively decomposed into smaller DFTs therefore reducing the number of computations.

2a)

$W\left[N\right]=\left[\frac{1}{2}-\frac{1}{4}\left(e^{\frac{j\left(2πn\right)}{m}}-e^{-\frac{j\left(2πn\right)}{m}}\right)\right]wR[n]$

$$\frac{1}{2}WR\left[n\right]-\frac{1}{4}WR[n](e^{j\left(w-\frac{2π}{m}\right)}+\frac{1}{4}WR[n](e^{j\left(w+\frac{2π}{m}\right)}$$

 2b.)

It is because the second and third terms widen the main lobe whereas they are lower side lobes than the rectangular window because they are lowered by the scaling factor.

2c.)

The width of the main lobe is known as the DFT sample points between two zero crossings and the highest level of side lobe which measures how many decibels down the highest side lobe is from the main lobe. It is important for the width of the main lobe to be narrow so that the same input signal can be seen with sharp edges and can be shifted to the right for better windowing.

3a.)

Y[n] = X[n]\*h[n] + X[-n]\*h[n]= X[n]\*hs[n]

$X\left[-n\right]\*h\left[n\right]= \sum\_{m=-\infty }^{\infty }X\left[-m\right]h\left[n-m\right]=\sum\_{m=-\infty }^{\infty }X\left[m\right]h\left[n+m\right]=X\left[n\right]\*h[-n]$

$Y\left[n\right]=X\left[n\right]\*h\left[n\right]+X\left[n\right]\*h\left[-n\right]=X\left[n\right]\left(h\left[n\right]+h\left[-n\right]\right)=>h\_{s}=h\left[n\right]+h[-n]$

*3b.)*

$H\_{s}\left(e^{jw}\right)= H\left(e^{jw}\right)+H\left(e^{-jw}\right)=2Re[H\left(e^{jw}\right)]$

*3c.)*

Yes it will hold because we can add delays to the filter, which will turn it from a non causal filter to a causal filter. The significance of this is that a program that already has all the data stored will not be affected by causality; therefore you can look further into the array because all the data has been gathered.