1)

1. The OS allocates processors by continuously and rapidly switching between different processes - waiting for hardware input, then executing instructions from a program, etc.
2. The OS abstracts out the hard drive by finding all the pieces of the relevant information and putting them together in memory. This way, the program only needs to worry about “opening” a “file”.
3. The OS abstracts out control over memory by mapping virtual addresses to physical addresses in memory, which prevents address conflicts and allows for data to be swapped in/out of storage as a means of expanding the available memory space.

2)  *30 years \* 12 months = 360 months; 360 months / 18 months per Moore’s Law cycle = 20 Moore’s Law cycles; 2^20 =* ***2,097,152x faster*** *- dang!*

3) *Show setups!*

1. 1/((1−0.75)+(0.75/2)) = 1/(0.25+0.375) = **1.6**
2. 1/((1−0.75)+(0.75/5)) = 1/(0.25+0.15) = **2.5**
3. 1/((1−0.75)+(0.75/500)) = 1/(0.25+0.0015) = **3.976**

4) A1, B1, B2, B3, A2, A3

5)

1. The actions are not idempotent because performing the instructions multiple times would fundamentally change the output/results.

|  |
| --- |
| Start transaction |
| Set Widget Quantity to 1100 |
| Change Widget UnitPrice to 1.15 |
| End transaction |

6) A transaction might need to be rolled back if the operation fails due to some kind of hardware problem (such as a broken storage drive or lack of sufficient storage space), or a software/procedural problem such as two transactions being stuck in a deadlock (which would necessitate cancelling one of those transactions).

7)

|  |
| --- |
| *SELECT Name, Role FROM employees WHERE Years\_employed >= 5;* |

8)

|  |
| --- |
| *SELECT \* FROM employees ORDER BY Name asc;* |

9)

1. **1,099**
2. Because that customer’s ID shows up twice, once for 948.5 and another for 150.5, for a total of 1,099.

10)

A.

|  |
| --- |
| *removeItem(colors,3);* |

B.

|  |
| --- |
| *colors[0]=”gold”;* |

C.

|  |
| --- |
| *console.log(colors[colors.length-2]);* |

^ The question asked for one line, but realistically I’d check the size of the array before referencing the index this way, in case the array was too small (or even empty)