**Level 1 (Presentation)**

**Level 2**

1. Multiple
   1. 1977
   2. 8-bit
   3. 8-bit = 255 binary = 11111111
   4. 16 bit
   5. 1048575 in decimals
2. Multiple
   1. 1978
   2. 65535 in decimal
   3. 20 bits
   4. 1048575 in decimal
3. Multiple
   1. 16 bit
   2. 16 bit
   3. 65535
   4. 24 bits
   5. 16777215
4. Multiple
   1. 64-bit processors have more capacity which means it can handle more things at once and better
   2. 32- bit processors are only able to use 4GB of ram while 64-bit processors can use more. Therefore the 64-bit processors will function better
   3. Since the 64-bit processor can run use more ram therefore function better faster.
5. Negative numbers are represented using bits by the Two’s complements in which it’s a mathematical operation which is used on binary numbers. The two’s complement is calculated by inverting the digits and adding one. Two complement is the most common method of representing signed integers on computers and more generally, fixed point binary values.
6. The term floating point is derived from the fact that there is no fixed number of digits before and after the decimal point; that is, the decimal point can float. There are also representations in which the number of digits before and after the decimalpoint is set, called fixed-point representations. Floating point representations are slower and less accurate than fixed point representations but can handle a larger range of numbers. These are stored in the computer memory by using four bytes which is 32 bits.

**Level 3**

1. Multiple
   1. bin(#) = prints number in binary
   2. oct(#) = prints number in octal
   3. hex(#) = prints number in hexadecimal
2. Multiple
   1. bin(‘11’) = Cannot turn a string into a binary number. Binary only works with numbers/integers
   2. int(‘11’) = Turns the string into integer with a value of eleven
   3. bin(int(‘11)) = Turns the string into an integer which is turned into binary.