**Level 1: Presentation Notes**

1. Number systems used in Computer Science
   1. List the main features of the Decimal System

* Digits: 2,1,2,3,4,5,6,7,8,9
* Used for communicating with human users
  1. List the main features of the Binary System(Base 2)
* Digits: 0,1(On or Off)
* Binary 10=Decimal 2
* Used by internal CPU & Memory Circuits  
  1. List the main features of the Octal System(Base 8)
* Digits:0,1,2,3,4,5,6,7
* Octal 10==decimal 8
* Used by computer scientists for grouping of 3 Binary Digits  
  1. List the main features of the Hexadecimal System(Base 16)
* Digits: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F(Uses extra letters)
* Hex F==Decimal 15
* Hex 10=Decimal 16
* Used by computer scientists for grouping of 4 binary digits

1. Compare and contrast the Decimal and Binary systems

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Decimal System** | **Binary System** |
| Digits  Used | 0,1,2,3,4,5,6,7,8,9 | 0,1(True, False)(+,-) |
| Addition Example | 0+1=1  1+1=2  99+1=100 | 0+1=1  1+1=10  11+1=100 |
| Powers of  Base | 100=1  101=10  102=100 | 20=1  21=10(or 2 decimal)  22=100(or 4 decimal) |
| Value of 111 | 111=102+101+100  (100+10+1) | 111=22+21+20  (Decimal= 4+2+1) |

1. Convert the following binary numbers to decimal:  
   1. 11 binary=3 decimal
   2. 101 binary=5 decimal
   3. 1010 binary=10 decimal
2. Convert the following decimal numbers to binary:  
   1. 6 decimal=110 binary
   2. 13 decimal=1101 binary
3. Add the following binary numbers. (verify your answers using decimal)

|  |  |
| --- | --- |
| 1. 0101(Decimal 5)   + 0010(Decimal 2)  = 0111 | 1. 0101(Decimal 5)   + 1010(Decimal 10)  = 1111 |
| 1. 0011(Decimal 3)   + 0010(Decimal 2)  = 0101 | d) 0110(Decimal 6)  + 0011(Decimal 3)  = 1001 |

1. List the main features of the following Computer Memory Structures:
   1. Bit

* 1 binary digit
* Used for Boolean data type
* Building Block for All computer data & memory  
  1. Byte
* 8 binary digits
* Largest Value: 1111 1111(28 – 1= 225 decimal)
* Used for char(character)(26 lower and upper case letters + 10 number symbols + punctuation marks + other stuff= about 130 distinct characters)  
  1. Word
* 16 binary digits(2 bytes)
* Largest value= 1111 1111 1111 1111(216-1=65,535 decimal)  
  1. Integer Data Type(short)
* Is one word(16 bits)
* Must represent both positive and negative
* Range: 32767 to -32768
* Larger or smaller numbers require a different data type  
    
  1. Double Word
* 32 binary digits( 4 bytes or 2 words)
* Largest value: 232-1= ~4 billion

**Level 2: Research Questions**

1. The Intel 8085 microprocessor was a first generation processor that was used in many early game systems and personal computers. Google “8085 microprocessor architecture” to answer these questions.
   1. Year Introduced

The processer was introduced by intel in 1976

* 1. Size of data bus (in bits)

The size of the data bus is 8-bit

* 1. Largest data number (in binary and decimal)

The largest 8 bit number is 255 and in binary it is 11111111.

* 1. Size of address bus (in bits)

The size of the address bus is 16-bit.

* 1. Largest memory address (in binary and decimal)

1048575 and in binary it is 1111111111111111.

1. The Intel 8086 microprocessor was the processor used in the first IBM PCs running the DOS operating system. Google “8086 microprocessor architecture” to answer these questions.
   1. Year Introduced

The processer was introduced by intel in 1978

* 1. Size of data bus (in bits)

The size of the data bus is 16-bit

* 1. Largest data number (in decimal)

The highest 16 bit number is 65535

* 1. Size of address bus (in bits)

The size of the address bus is 20-bit.

* 1. Largest memory address (in decimal)

The highest 20 bit number is 1048575.

1. The Intel 80286 microprocessor a common processor used in IBM PCs running the Windows operating system. Google “80286 microprocessor architecture” to answer these questions.
   1. Year Introduced

The processer was introduced by intel in 1982

* 1. Size of data bus (in bits)

The size of the data bus is 16-bit

* 1. Largest data number (in decimal)

The highest 16 bit number is 65535

* 1. Size of address bus (in bits)

The size of the address bus is 24-bit.

* 1. Largest memory address (in decimal)

The highest 24 bit value is 16777215

4. The modern PCs run either a 32 bit or 64 bit Windows operating system. Google “32 vs 64 bit” to answer these questions.

1. How do these systems differ in data capacity? (explain using bits)

On a 32 bit computer, you can have a max of 2TB hard drive disk. On a 64 bit computer, you can have 16 TB.

1. How do these systems differ in memory capacity? (explain using bits)

The 32-bit computers can handle 4GB maximum usable memory, or 232 bytes. The 64-bit can handle much more, such as 8GB, 16GB, and 32GB.

1. How do these systems differ in hardware requirements?

They may need minimum RAM requirements or minimum storage requirements. A processor minimum requirement may also affect how the computer runs.

1. Research and explain how negative (-) numbers are represented using bits and how they are stored in computer memory.  
   The number 1 is negative and the number 0 is positive. They are stored as 0’s and 1’s. An example will be “01001011101010100010010101111010001010111100101111”. They are stored into 8 bits (1 byte). If a value contains more data that can fit into a single byte, it is stored using multiple bytes.
2. Research and explain how floating point (decimal) numbers are represented using bits and how they are stored in computer memory.

Floating-point numbers are encoded by storing the significand and the exponent. 8 digits are used to represent a floating point number.

**Level 3: Sample Program**

1. Modify the following sample Python program to print out the digits in:

number = input("Enter a 4 digit decimal number:")

index = 0

for char in number :

index += 1

print("Digit ", index, " is : ", char)

* 1. Binary

nDecimal = eval(input("Enter a positive or negative number:"))

print("Entered Number:", nDecimal)

nbin=[]

while nDecimal > 0 or nDecimal < 0:

value = int(nDecimal % 2)

nDecimal = int(nDecimal / 2)

nbin.append(value)

nbin.reverse()

print("Your binary number is", end=": ")

for x in nbin:

print(x, end='')

* 1. Octal

number=int(input("Enter a number to convert into Octal:"))

print(oct(number))

* 1. Hexadecimal

number=int(input("Enter a number to convert into Hexadecimal:"))

print(hex(number))