**Level 1: Presentation Notes**

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

* Digits: 1,2,3,4,5,6,7,8,9
* Used for communicating with human users  
  1. List the main features of the Binary System
* Digits: 0,1 (On or Off)
* Used by internal CPU and Memory circuits  
  1. List the main features of the Octal System
* Digits:01,1,2,3,4,5,6,7, (No digits 8 and 9)
* Octal 10 == Decimal 8
* Used by computer scientists for grouping of 3 binary digits  
  1. List the main features of the Hexadecimal System
* 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 or False)  (+,-) |
| Addition Example | 0+1=1  1+1=2  9+1=10 | 0+1=1  1+1=12  11+1=10 |
| Powers of  Base | 100=1  101=10  102=100  etc. | 20=1  21=10 (or 2 decimal)  22=100 (or 4 decimal)  etc. |
| Value of 111 | 111=102+101+100  (100+10+1) | 111=22+21+20  (Decimal: 4+2+1=7) |

1. Convert the following binary numbers to decimal:  
   1. Decimal 11 = 3 Decimal: 103 102 101 100  
       1000 100 10 1
   2. Decimal 101 = 5  
       Binary: 23 22 21 20
   3. Decimal 1010 = 10 8 4 2 1
2. Convert the following decimal numbers to binary:  
   1. Binary 6 = 0110
   2. Binary 13 =1101
3. Add the following binary numbers. (verify your answers using decimal)

|  |  |
| --- | --- |
| a) 0101 Decimal(5)  + 0010 + Decimal(2)  111 7  4+2+1=7 | b) 0101 Decimal(5)  + 1010 + Decimal(10)  1111 15  8+4+2+1=15 |
| c) 0011 Decimal(3)  + 0010 + Decimal(2)  101 5  4+1=5  Put a zero in the column and carry the one over to other side. | d) 0110 Decimal(6)  + 0011 + Decimal(3)  1001 9  8+1=9  Put a zero in the column and carry the one over to other side. |

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 and memory

* 1. Byte

•8 binary digits

•Largest value: 1111 1111 (28 – 1 = 255 Decimal)

•Used for Char (character) data type

•26 lower case letters + 26 uppercase letters

+ 10 number symbols + punctuation marks+ Other Stuff

equal about 130 distinct characters

* 1. Word

Short: •16 binary digits (2 bytes)

Largest value: 1111 1111 1111 1111

(216 – 1 = 65,535 Decimal)

Long:•Provides much larger range than Integer for

Positive (+) and Negative (-) numbers

* 1. Integer Data Type

•Is 1 Word (16 bits)

•But 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 approx)

•Double Word Memory Addressing

•Provides access to about 4 GB of memory max.

**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

* It was introduced in 1967
  1. Size of data bus (in bits)
* It has an 8 bit data bus
  1. Largest data number (in binary and decimal)
* The largest data number in binary is 1111 1111 and in decimal is 28-1=255
  1. Size of address bus (in bits)
* It has a 16 bit address bus with
  1. Largest memory address (in binary and decimal)
* The largest address number is 1111 1111 1111 1111 in binary and 216-1=65535 in decimal

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

* It was introduced in 1976
  1. Size of data bus (in bits)
* It has a 16 bit data bus
  1. Largest data number (in decimal)
* The largest data number in binary is 1111 1111 1111 1111 and in decimalis 216-1=65535 in decimal
  1. Size of address bus (in bits)
* It has a 20 bit address bus
  1. Largest memory address (in decimal)
* The largest address memory in binary is 1111 1111 1111 1111 1111 and in decimal is220-1=1048575

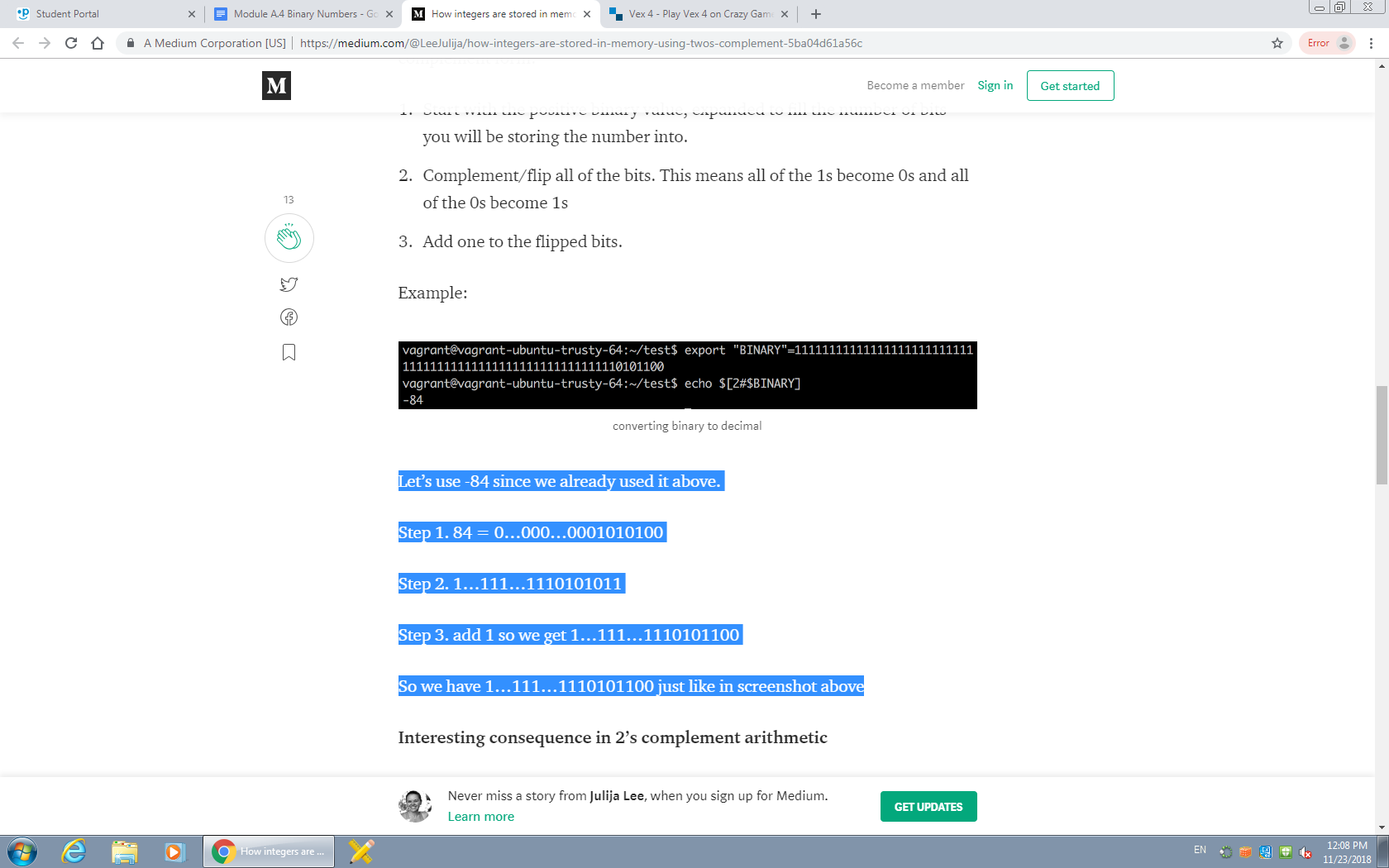
1. The Intel 80286 microprocessor a common processor used in IBM PCs running the Windows operating system. Google “80286 microprocessor architecture” to answer these questions.
   1. Year Introduced

* It was introduced in 1982
  1. Size of data bus (in bits)
* It has a 16 bit data bus
  1. Largest data number (in decimal)
* The largest data number in binary is 1111 1111 1111 1111 and in decimalis 216-1=65535 in decimal
  1. Size of address bus (in bits)
* It has a 24 bit address bus
  1. Largest memory address (in decimal)
* The largest address memory in binary is 1111 1111 1111 1111 1111 1111 and in decimal is 224-1=16777215

1. 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)

* 32 bit processors are capable of handling 4GB while 64 bit processors are capable of handling more than 4GB
* the 32 bit data capacity is 232-1=4,294,967,295 in decimal and 32 bits
* the 64 bit data capacity is 262-1= 18,446,744,073,709, 551,615 and 64 bits
  1. How do these systems differ in memory capacity? (explain using bits)
* 64 bit processor is capable of storing more computational values, including memory addresses it means that about 4 billion times of 32 bit processor
* the 32 bit data capacity is 232-1=4,294,967,295 in decimal and 32 bits
* the 64 bit data capacity is 262-1= 18,446,744,073,709, 551,615 and 64 bits
  1. How do these systems differ in hardware requirements?
* 64 bit processors aren’t capable of running 16 bit programs because they are older and sometimes aren’t capable of running older 32 bit programs
* 32 bit processor can’t install a 64 bit operating system, but it can install a 32 bit operating system

1. Research and explain how negative (-) numbers are represented using bits and how they are stored in computer memory.

* there are compiler differentiates between positive and negative number from a bit
* MSB (most significant bit) is used in the process of this
* if there’s a 0 the number is positive although if there’s a 1 the number is negative
* to convert a negative number into a number that matches 2’s complement form you first need to start with a positive binary value expanded to fit the bits in the system
* then you flip the bits which means all the 1’s become 0’s and all the 0’s become 1’s
* after that you add one
* an example:

Let’s use 84 bits

Step 1. 84 = 0…000…0001010100

Step 2. 1…111…1110101011

Step 3. add 1 so we get 1…111…1110101100

So we have 1…111…1110101100 just like in screenshot above

1. Research and explain how floating point (decimal) numbers are represented using bits and how they are stored in computer memory.

* The representation of unsigned binary fractions proceeds in exactly the same way as decimal fractions. For example

0.62510 = 1\*0.5 + 0\*0.25 + 1\*0.125 = 1\* 2-1 + 0\* 2-2 + 1\* 2-3 = 0.1012

* to the right of each decimal point represents a negative power of 2, like how decimals represent a negative power of 10. Likewise,
* if there are m bits to the right of a decimal, the precision of the number is 2-m
* These are commonly used by Digital Signal Processors.

**Level 3: Sample Program**

1.    Explain the result of the following Ptyhon operations:

a.    bin(11)

The number 11 in printed in binary.

b.    oct(11)

The number 11 in printed in Octal.

c.    hex(11)

The number 11 in printed in Hexadecimal.

2.    Explain the following Ptyhon operations:

a.    bin(‘11’)  - Why does this operation give an error?

It gives an error because the operation is trying to turn a string into a binary number and only integers can be turned into binary.

b.    int(‘11’)   - Why does this work?

This works as it turns the string into an integer.

c.    bin(int(‘11’))   - Why does this fix the problem?

This fixes the problem because the operation first turns the string into an integer and this means that 11 can be turned into a binary number.

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

a.    Binary

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | number = input("Enter a 4 digit decimal number:")  index = 0  for char in number :    index += 1    Binary = bin(int(number))    digit = bin(int(char))    print("The number ", char, " in Binary is : ",digit)    print('The number', number,'in Binary is', Binary) |

b.    Octal

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | number = input("Enter a 4 digit decimal number:")  index = 0  for char in number :    index += 1    Oct = oct(int(number))    digit = oct(int(char))    print("The number ", char, " in Octal is : ",digit)    print('The number', number,'Octal is', Oct,'\n') |

c.    Hexadecimal

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | number = input("Enter a 4 digit decimal number:")  index = 0  for char in number :    index += 1    hexa = hex(int(number))    digit = hex(int(char))    print("The number ", char, " in hexadecimal is : ",digit)    print('The number', number,'in hexadecimal is', hexa,'\n') |

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

index = 0

for char in number :

  index += 1

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