Namaste!, my name is Sandeep

M. Kadam, Will be presenting before you the module coding

schemes from the topic computer fundamentals.

In this module we are going to cover concept of coding schemes,

ASCII and Unicode coding systems and representation of characters

in ASCII and Unicode.

At the end of this module you will be able to understand the

concept of coding system,

explain it's need,

cite the names of various coding

systems. Remember and write the codes for various

characters in ASCII and Unicode,and finally compare ASCII and

Unicode coding systems.

Question comes what are coding

schemes? They are nothing but the standards for representing

characters in computers.

They are required for the transfer or an exchange of data.

Question comes between what? Between the computers and

between computers and peripheral devices connected to the

computers. Also they enable,

for the independent development of several

hardware and software components for a wide variety

of computers. As you know that coding system is required

everywhere, every human being is using the coding system.

But we are restricting our coding system to the computer

coding system itself.

How the characters are represented in the computers? by

means of bits , that's zeros and ones. You know that bit refers

to a binary digit, is a short form for binary digit and we know

that computers understand zeros and ones, that you have learnt

very well in the previous module. Hence computers use binary coding

schemes to represent data internally. A group of bits used

to represent a character is called as a byte. Normally, a

byte refers to 8 bits. In other words, one character consumes

one byte of computer's memory.

Interestingly, externally it appears that human beings use

natural language symbols to communicate with computers,

but internally computers convert everything into binary

data and then process all information in binary word.

Finally, computer converts binary information into human

understandable language.We have different types of

coding systems, BCD,

EBCDIC, ASCII, Unicode, ISCII and so on, but most

commonly used ones are ASCII and

Unicode. ASCII has been introduced by ANSI in the year

1963 and subsequently revised in 1960, 1968, 1977 and finally the

most recent update in the year

1986. Is commonly used in the transmission of data

through data communications and also used almost

exclusively to represent data internally in

micro computers or some micro computers which have

been designed using ASCII coding system itself.

There are two types of ASCII.

ASCII-7 and ASCII-8. ASCII-7 uses 7 bit combination and hence

supports 128 characters. Where as ASCII-8 supports,

256 characters by the combination of eight bits.

The codes of first 128

characters for both the ASCII-7 and ASCII-8

are identical. Let us consider the ASCII table.

Where binary representation of each and every character is

represented. If you look at the characters over here, namely

digits. Digits are represented in ASCII codes- 0110000,0110001...

and you find here one peculiar pattern for this. Entire code

has been divided into two parts. One is numeric portion, the

second one is, the first one is Zone bit portion and the second

one is numeric bit portion and for all the digits zone bit

portion is common whereas the numeric portion is binary equivalent

to digits. And this entire binary is converted into decimals,

48, 49, 50,51 and so and so forth, whereas in Hexadec symbol

the same binary is converted as 30,31,32,33 and so on.

Let us consider here the alphabets, capital alphabets.

Capital Alphabet A is represented as 1000001. You see the zone bit

portion which is common for all the alphabets to certain numbers

and then the equivalent decimal is 65, 66, 67 and so on.

Whereas the same Hexadecimal conversion is 41,42.43,

and 44.

Let us look at the next or continuation. You find the

same continuation pattern over here, and similarly for the

alphabets the pattern over here is , zone bit portion is same,

whereas for the small alphabets it is different

zone bit portion 110. You can see over here and the rest

one is the binary equivalent to the numbers.

In the previous table we have seen some ASCII-7 codes for

digits 0 to 9, are 011000 to 00111001 in binaries

also 31- hexadecimal to 39-

hexadecimal. Similarly, wefind for uppercase Alphabets

A to Z, we find this representation and for lower

alphabets a to z, we find 1100001 to 1111010 in Binaries

also 61 hexadecimal to 7A hexadecimal.

It is worth to note that ASCII uses exactly similar

number system as it's four-to-one shortcut notation for

memory dump in computers.

It is very interesting to see how these characters look inside

the computer. Let us take the word COLD and let us

understand how exactly it looks inside the computer.

You have noted from the previous table, C refers to this binary with 43

in hexadecimal, 67 in decimal. Then we have 0, Sorry,

then we have O, then we have L and then we have D. Let us see

how this COLD, when it is entered in the computer, how it

looks in binary? See, this is the way how it represents in the

binary. That means computer understands this as a pattern

of zeros and ones.

Whereas in hexadecimal the same code is referred as 43, 4F, 4C

and 44. From above, It is clear that the word COLD

consumes 4 bytes of the computer's memory using ASCII-7.

However, there are limitations

to ASCII. It supports only 128 characters. Limited to English

language only. Do not have enough number of bits to

accommodate language specific symbols of different languages

other than English.

What is UNICODE?

Is an encoding system that provides a unique number for

every character, no matter what is the problem? No matter what.

are the programs? No matter what is the language? This is as per

the official website of Unicode.

Why it is required?

It is required since there is no

single Coding System which could represent vast number

of characters in different

languages. Many encoding systems conflicted with one another.

Difficulty in exchange of text files at international level.

Let us look at the brief history.

The Unicode Consortium, which is a nonprofit organization,

coordinates the development of Unicode. However there

are many big companies who are the members of

this consortium.

The first volume of the Unicode Standard was published in 1991 and as of March 2020 one can see Unicode Standard Version 13.0. A repertoire of 143,859 characters, covering of 154 modern and historic scripts, multiple symbol sets and emoji. Let us see some of the features of Unicode. There is a provision of consistent way of encoding multilingual plain text. Defines codes for characters used in all major languages across the world. Defines codes for special characters. Capacity to encode as many as million characters. Assigns each character, a unique numeric value and a name. Reserves a part of the code space for private use to enable users to assign codes for their own characters and symbols. Affords simplicity and consistency of ASCII, and finally specifies an algorithm for presentation of text with bi-directional behavior. We find three different formats of unicode coding system, namely UTF-8, UTF-16, UTF-32. In UTF-8 it is a byte oriented format, have the same byte values as that of ASCII. Characters are represented as variable length encoding of one, two, three or four bytes. Very

popular for HTML and similar

protocols. UTF 16 is a Word oriented format. As you are

aware about, one word is equal to 16 bits and characters are

represented as variable length encoding of one or two words.

And finally we have UTF-32,

which is a double word oriented format with fixed length of 32

bits. Unicode characters are represented as fixed length

encoding of two words. Let us have a look at a table which

represents a character code, for UTF-16.

And you can find here the

characters, A,B,C,D and then its corresponding binaries will find

it in the 16 bit, which is exactly, the last two parts are

exactly similar to that of ASCII-7

or ASCII-8 and with hexadecimal code representing

U+0041 and so and so forth. And decimal 65,66, 67

and similarly for these also, the last part is

similar to that of the ASCII.

Unicode uses exactly similar number system as it's four-to-

one shortcut notation for memory dumping computers. The word

COLD will look like in Hexadecimal as follows.

U+ 0043, U+004F, U+004C, U+ 0044 and

Interestingly, the same pattern you find in binaries

represented like these long patterns of zeros and ones.

Having learnt both the ASCII coding system and UNICODE coding

system, let us have a quick comparison. ASCII supports only

128 to 256 characters. Where as Unicode supports more than 11

lakhs characters. ASCII supports only English language,

whereas UNICODE supports every possible language. ASCII

supports limited special characters, UNICODE supports

large number of special characters. ASCII consumes less

memory, UNICODE consumes more

memory. In ASCII, 7 bit coding formula is used, whereas UNICODE

uses 8, 16 and 32 bit encoding systems. And finally ASCII

enables software product to be designed for English language

only, whereas UNICODE enables single software product or a

single website to be designed for multiple platforms,

languages and countries(no need for re-engineering). To summarize,

in this module we have learnt the definition of coding schemes,

need for coding schemes, representation of characters in

computers, types of coding

systems. We have learnt the binary representation of

characters using ASCII as well as its limitations.

The need and features of UNICODE, binary representation of characters using UNICODE, and finally we have compared ASCII coding system and UNICODE coding systems. For more information you can have the following references. I hope you have enjoyed this session. Thank you very much.