CLASS : XI SCIENCE
SUBJECT : COMPUTER SCIENCE – PYTHON (083)
NOTE:
***FOR NOW STUDENTS NEED NOT WORRY ABOUT THE TEXT BOOK OR REGISTER.**
***KINDLY GO THROUGH THE NOTES .**
***THE NOTES SHOULD BE WRITTEN IN ANY ROUGH REGISTER UNDER THE HEADING**

UNIT1 Computer Systems and Organisation

- Basic computer organisation: description of a computer system and mobile system, CPU, memory, hard disk, I/O, battery.
- Types of software: application, System, utility.
- Memory Units: bit, byte, MB, GB, TB, and PB.
- Boolean logic: OR, AND, NAND, NOR, XOR, NOT, truth tables, De Morgan's laws
- Information representation: numbers in base 2, 8, 16, binary addition
- Strings: ASCII, UTF8, UTF32, ISCII (Indian script code), Unicode
- Basic concepts of Flowchart
- Concept of Compiler & Interpreter
- Running a program: Notion of an operating system, how an operating system runs a program, idea of loading, operating system as a resource manager.
- Concept of cloud computing, cloud (public/private), introduction to parallel computing.

Data Representation: It refers to the internal method used to represent various types of data stored on a computer. Computers use different types of numeric codes to represent various forms of *data*, such as text, number, graphics and sound.
To know data representation in computer we must know following number system

Decimal number system (Base=10):- 0,1,2,3,4,5,6,7,8,9.
Binary number system (Base=2):- 0,1.
Octal number System (Base=8):- 0,1,2,3,4,5,6,7.
Hexa Decimal System(Base =16):- 0,1,2,3,4,5,6,7,8,9,A(10),B(11),C(12),D(13),E(14),F(15).
A repeated division and remainder algorithm can convert decimal to binary, octal, or hexadecimal.
Divide the decimal number by the desired target radix (2, 8, or 16).
Append the remainder as the next most significant digit.
Repeat until the decimal number has reached zero.

Character	ASCII code	Decimal Equivalent
g	01100111	103
r	01110010	114
a	01100001	97
d	01100100	100
e	01100101	101
u	01110101	117
p	01110000	112

Decimal to Binary:

The decimal (base ten) numeral system has ten possible values (0,1,2,3,4,5,6,7,8, or 9) for each place-value. In contrast, the binary (base two) numeral system has two possible values represented as 0 or 1 for each place-value.

•Example of using repeated division to convert 1792 decimal to binary:

Decimal Number	Operation	Quotient	Remainder	Binary Result
1792	÷ 2 =	896	0	0
896	÷ 2 =	448	0	00
448	÷ 2 =	224	0	000
224	÷ 2 =	112	0	0000
112	÷ 2 =	56	0	00000
56	÷ 2 =	28	0	000000
28	÷ 2 =	14	0	0000000
14	÷ 2 =	7	0	00000000
7	÷ 2 =	3	1	100000000
3	÷ 2 =	1	1	1100000000
1	÷ 2 =	0	1	11100000000
0	done.			

Decimal to Octal:

Example of using repeated division to convert 1792 decimal to octal:

Decimal Number	Operation	Quotient	Remainder	Octal Result
1792	÷ 8 =	224	0	0
224	÷ 8 =	28	0	00
28	÷ 8 =	3	4	400
3	÷ 8 =	0	3	3400
0	done			

Decimal to Hexadecimal

Example of using repeated division to convert 1792 decimal to hexadecimal:

Decimal Number	Operation	Quotient	Remainder	Hexadecimal Result
1792	÷ 16 =	112	0	0
112	÷ 16 =	7	0	00
7	÷ 16 =	0	7	700
0	done.			

48879 decimal converted to hex is:

Decimal	Operation	Quotient	Remainder	Hexadecimal Result
48879	÷ 16 =	3054	15	F
3054	÷ 16 =	190	14	EF
190	÷ 16 =	11	14	EEF
11	÷ 16 =	0	11	BEEF
0	done			

Octal To Binary

Converting from octal to binary is as easy as converting from binary to octal. Simply look up each octal digit to obtain the equivalent group of three binary digits.

Octal:	0	1	2	3	4	5	6	7
Binary:	000	001	010	011	100	101	110	111

Octal to Hexadecimal

When converting from octal to hexadecimal, it is often easier to first convert the octal number into binary and then from binary into hexadecimal. For example, to convert 345 octal into hex:

[According to previous example]

Octal =	3	4	5	
Binary	011	100	101	011100101 binary

Drop any leading zeros or pad with leading zeros to get groups of four binary digits (bits):

Binary 011100101 = 1110 0101

Then, look up the groups in a table to convert to hexadecimal digits.

Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	0	1	2	3	4	5	6	7
Binary:	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal	8	9	A	B	C	D	E	F

Octal to Decimal

Converting octal to decimal can be done with repeated division.

1. **Start the decimal result at 0.**
2. **Remove the most significant octal digit (leftmost) and add it to the result.**
3. **If all octal digits have been removed, you're done. Stop.**
4. **Otherwise, multiply the result by 8.**
5. **Go to step 2.**

Octal Digits	Operation	Decimal Result	Operation	Decimal Result
345	+3	3	× 8	24
45	+4	28	× 8	224
5	+5	229	done.	

The conversion can also be performed in the conventional mathematical way, by showing each digit place as an increasing power of 8.

345 octal = $(3 * 8^2) + (4 * 8^1) + (5 * 8^0) = (3 * 64) + (4 * 8) + (5 * 1) = 229$ decimal

Converting from hexadecimal is next.

Converting from hexadecimal to binary is as easy as converting from binary to hexadecimal. Simply look up each hexadecimal digit to obtain the equivalent group of four binary digits.

Hexadecimal to Octal

When converting from hexadecimal to octal, it is often easier to first convert the hexadecimal number into binary and then from binary into octal. For example, to convert A2DE hex into octal:

(from the previous example)

Add leading zeros or remove leading zeros to group into sets of three binary digits.

Binary: 1010001011011110 = 001 010 001 011 011 110

Then, look up each group in a table:

Binary:	000	001	010	011	100	101	110	111
Octal:	0	1	2	3	4	5	6	7

Therefore, through a two-step conversion process, hexadecimal A2DE equals binary 1010001011011110 equals octal 121336.

Hexadecimal to Decimal

Converting hexadecimal to decimal can be performed in the conventional mathematical way, by showing each digit place as an increasing power of 16. Of course, hexadecimal letter values need to be converted to decimal values before performing the math.

Hexadecimal:	0	1	2	3	4	5	6	7
Decimal:	0	1	2	3	4	5	6	7
Hexadecimal:	8	9	A	B	C	D	E	F
Decimal	8	9	10	11	12	13	14	15

A2DEhexadecimal:

$= ((A) * 16^3) + (2 * 16^2) + ((D) * 16^1) + ((E) * 16^0)$

$= (10 * 16^3) + (2 * 16^2) + (13 * 16^1) + (14 * 16^0)$

= (10 * 4096) + (2 * 256) + (13 * 16) + (14 * 1)

= 40960 + 512 + 208 + 14

= 41694 decimal

Arithmetic operation in Binary Systems

Addition

Rules for carrying out binary Additions are :

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 with one (1) carry over.

Example

1. For adding 101110_2 and 111101_2

Thus 1101011 in binary system is equivalent to 107 in decimal system.

EXAMPLE 2.10 *Add the binary numbers 01010111 and 00110101*

Solution. If you add the bits column by column as earli
explained you will get

```
 1 1 1   1 1 1     ← Carries
  0 1 0 1 0 1 1 1
+ 0 0 1 1 0 1 0 1
 1 0 0 0 1 1 0 0
```

EXAMPLE 2.11 *Add the binary numbers 1011 and 110.*

Solution.

```
 1 1       ← Carries
  1 0 1 1
+   1 1 0
1 0 0 0 1
```

EXAMPLE 2.12 *Add binary numbers 11110 and 11.*

Solution.

```
 1 1 1       ← Carries
  1 1 1 1 0
        1 1
1 0 0 0 0 1
```

CHARACTER / STRING REPRESENTATION –

Character representation means representing alphabets(upper as well as lower case) , digits (0 -9) special symbols, non printable characters etc.
ASCII CODE : Pronounced ask-ee, ASCII is the acronym for the American Standard Code for Information Interchange. It is a code for representing 128 English characters as numbers, with each letter assigned a number from 0 to 127. For example, the ASCII code for uppercase M is 77.

EXAMPLE 2.15 *An operator is typing in a BASIC program at the keyboard of a certain micro-computer. The computer converts each keystroke into its ASCII code and stores the code in memory. Determine the codes that will be entered into memory when the operator types in the following BASIC statement :*

GOTO 25

Solution. Locate each character (including the space) in Table 2.7 and record its ASCII code.

G	1000111
O	1001111
T	1010100
O	1001111
(Space)	0100000
2	0110010
5	0110101

The main advantage of ASCII is its simplicity — it uses one byte to represent one character. There is extended ASCII that uses 8 bits to represent various characters. It can represent 256 characters as opposed to 128 characters of ASCII.

...om ASCII there are other systems that are also used to represent various symbols. ...following lines, we are talking about some of these — *ISCII* and *Unicode*.

The ASCII table is divided into three different sections.

- **Non-printable, system codes between 0 and 31.**
- **Lower ASCII, between 32 and 127. This table originates from the older, American systems, which worked on 7-bit character tables.**
- **Higher ASCII, between 128 and 255. This portion is programmable; characters are based on the language of your operating system or program you are using. Foreign letters are also placed in this section.**

Unicode Transformation Format UTF-8 is a compromise character encoding that can be as compact as ASCII (if the file is just plain English text) but can also contain any Unicode characters (with some increase in file size). UTF stands for Unicode Transformation Format. The '8' means it uses 8-bit blocks to represent a character.

UTF-32 (32-bit Unicode Transformation Format) is a fixed-length encoding used to encode Unicode code points that uses exactly 32 bits (four bytes) per code point (but a number of leading bits must be zero as there are far fewer than 2^{32} Unicode code points).[*citation needed*] **UTF-32 is a fixed-length encoding, in contrast to all other Unicode transformation formats, which are variable-length encodings. Each 32-bit value in UTF-32 represents one Unicode code point and is exactly equal to that code point's numerical value.**

ISCII Utilities are programs for analyzing text files encoded according to the Indian Script Code for Information Interchange (ISCII), the Indian national standard. IsciiName identifies each code, printing the byte offset, the code in hex, and an explanation of the meaning of the code.

Unicode. Unicode is a universal character encoding standard. It defines the way individual characters are represented in text files, web pages, and other types of documents. ... While ASCII only uses one byte to represent each character, Unicode supports up to 4 bytes for each character.

****