CIT 101: INTRODUCTION TO COMPUTER SCIENCE

FIRST SEMESTER 2020/2021 ACADEMIC SESSION

Outline

- Data types and organization:

Definition of terms like bit, byte, ASCII,

- EBCDIC, UNICODE, Number systems, Data coding, Binary arithmetic, Data Representation(fixed and floating point).

Preamble

- A fundamental course in computer science which enables students to have the much needed foundation and appreciate the areas the computer system represents numbers in its memory using its techniques and applications.

- Helps in understanding the concepts of data and its type

- Distraction avoided in order to have sufficient knowledge in understanding the basic arithmetic operations and how it is performed arithmetically for easy comprehension of the applicati ons.

- course taught to appreciate but in reality it helps one to learn the fundamental of computer science in the course in order to appreciate the course at large.

Introduction of Basic concepts

Data

- They are values or set of values. Also refer to facts, events, activities and transactions which have been recorded. It is the raw material from which information is produced. It could be listed, stored, up-dated, some or all of it removed or merged.

Data Processing

- Describes how data is converted into information.

Data Type

- It is a way to classify various types of data and values which varies in a programming language. Thus are Simple, Structured and pointer data types and referred to as primitive data types

Data Object

- It represents a list of elements. For example a list of integers or list of alphabetical strings as data objects

BASIC CONCEPTS CONTD.

- **Data Item:-** This refers to single unit of values.
- **Group Items:-** Data item that are divided into sub items.
- **Elementary Items:-** Data item that cannot be divided.
- **Attribute and Entity:-** An entity is that which contains certain attributes or properties which may be assigned values.
- **Entity Set:-** Entities of similar attributes form an entity.
- **Field:-** Field is a single elementary unit of information representing an attribute of an entity.
- **Record:-** It is a collection of field values of a given entity set.
- **File:-** It is a collection of the entities in a given entity set

How is Data processed

- Data can be listed, stored, up-dated and some or all of it can be removed or merged together.

Some ways to process data into information are as follows:

- Calculating: This is a process by performing arithmetic manipulation such as adding, subtracting, dividing and multiplying
- Sorting: it is a process by arranging data into a predetermined order or sequence. For example a class attendance may be arranged in the order of students' names or in the order of students' numbers. Sometimes from small or low values to bigger or higher values

How is Data processed Contd

- Classifying

This is a process of identifying certain characteristics in an item of data and putting them into a certain category according to those characteristics. Items of data with the same characteristics can be grouped together. For example you can classify the students in this class by the following the ways:

- Age
- Gender (Male, Female)
- Course programme

- Summarizing: This is a process of processing data whose purpose is to highlight or condense certain information so as to draw it more easily or quickly to the attention of the person who will use the information, for example the contents of a long report may be summarized into a few short paragraphs.

Characteristics of Data

- Atomic: It should define a single concept.
- Traceable: It should be able to map out some data element.
- Accurate: It should be unambiguous.
- Clear and Concise: It should be understandable.

Types of Data Types

- The data types are

1. **Simple/basic data type**

They are simple items such as numbers, characters which are associated with single identifiers on a one to one basis. For example, integers, constants, variables, expressions, functions, Boolean (True/False), floating (Decimal numbers), Character and strings. The data type integer consists of the objects {0, +1, -1, +2, -2, …, INT_MAX, INT_MIN} and the operations +, -, *, /, and %. They are also known as built-in data type.

2. **Structured/group data type**

This consist of multiple data items that are related to one another in some specified manner. Each group of data items is associated with a particular identifier and also with a corresponding individual identifiers. Examples include arrays, records, files and sets. This is known as derived data type

3. **Pointer data type**

They are used to construct dynamic structured data types.

4. **The user-defined type**

DATA REPRESENTATION AND COMPUTER ARITHMETIC

- The main features are as follows
- Non Positional Number System
- Positional Number System
- Number Systems
- Conversion
- Shortcut Methods
- Every computer stores numbers, letters, and other special characters in coded form. Before going into the details of these codes, it is essential to have basic understanding of some terms that will be essential in this area. Hence, it will familiarizes you with the fundamentals of number system. It also introduces some commonly used number system by computer professionals and relationship among them.
- Number systems are two types
- non-positional
- positional.

NON POSITIONAL NUMBER SYSTEM

- In early days, human beings counted on fingers. When counting beyond ten fingers, they used stones , pebbles, or sticks to indicate values.
- This method of a counting uses an additive approach or non-positional number system.
- In this system, we have symbols such as I for 1, II for 2, III for 3, IIII for 4, IIIII for 5, etc.
- Each symbol represents the same value regardless of its position in a number, and to find the value of a number, one has to count the number of symbols present in the number.
- Since it is very difficult to perform arithmetic with such a number system, positional number system was developed.

POSITIONAL NUMBER SYSTEM

- In a positional number system, there are only a few symbols called digits. These symbols represent different values, depending on the position they occupy in a number. The value of each digit in such a number is determined by three considerations.
- The digit itself,
- The position of the digit in the number, and
- The base of the number system (where base is defined as the total number of digits available in the number system).
- In our day-to-day life, we use decimal number system. In this system, base is equal to 10 because there are altogether ten symbols or digit (0,1,2,3,4,5,6,7,8, and 9).
- You know that in decimal number system. Successive positions to the left of the decimal point represent units, tens, hundreds, thousands, etc. However, notice that each position represents a specific power of the base (10).
- For example, decimal number 2586 (written as 2586) consists of digit 6 in units position, 8 in tens position, 5 in hundreds position, and 2 in thousands position, and its value can be written as:
- $(2 \times 10^3) + (5 \times 10^2) + (8 \times 10^1) + (6 \times 10^0) = 2000 + 500 + 80 + 6 = 2586$

- Observe that the same digit signifies different values, depending on the position it occupies in the number. For example,
-
- In 2586_{10} the digit 6 signifies $6 \times 10^0 = 6$
- In 2568_{10} the digit 6 signifies $6 \times 10^1 = 60$
- In 2658_{10} the digit 6 signifies $6 \times 10^2 = 600$
- In 6258_{10} the digit 6 signifies $6 \times 10^3 = 6000$
- Hence, we can represent any number by using the available digits and arranging them in various positions.
- The principles that apply to decimal number system, also apply to any other positional number system. It is important to keep track of only the base of the number system in which we are working.
- The value of the base in all positional number systems suggests the following characteristics:
- The value of the base determines the total number of different symbols or digits

  available in the number system. The first of these choices is always zero.
- The maximum value of a single digit is always equal to one less than the value of the base.

Binary Number System

- Binary number system is like decimal number system, except that the base is 2, instead of 10.
- We can use only two symbols or digits (0 and 1) in this number system.
- Note that the largest single digit is 1 (one less than the base).
- Each position in binary number represents a power of the base (2).
- Hence, in this system, the rightmost position is units (2^0) position,
- the second position from the rights 2's (2^1) position
- proceeding in this way, we have 4's (2^2) position
- 8's (2^3) position, 16's (2^4) position, and so on.
- Therefore, decimal equivalent of binary number 10101 (written as 10101) is : 2
- $(1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 16 + 0 + 4 + 0 + 1 = 21$
- In order to be specific about which system we are referring to,
- it is a common practice to indicate the base as a subscript.
- Hence, we write" $10101_2 = 21_{10}$

- The sort form of **“binary digit” is bit**.
- Hence, “bit” in computer terminology means binary number system that has either a 0 or 1 as its digits.
- Also digital computers uses the binary system which is convenient because it can easily be related to ON and OFF situations.
- **1 is equivalent to ON while 0 is equivalent to OFF**
- Combinations of these ONs and OFFs relay specifically send messages to the computer
- To remove confusion and make retrieval of information easier, the computer uses coding schemes to represent characters
- Four-bit Binary coded Decimal (4-bit BCD) is one of the coding schemes which uses 4-bits to represent each decimal digit in a number.
- For example to represent the number 32, there will be two groups of four bits each which are as follows:

➢ One group will represent “3”

➢ Another group will represent “2”

The 4-bit BCD saves space when large decimal numbers are to be represented and is also easier to convert 4-bit BCD back to its decimal number than to convert the long binary representation to decimal.

- It allows sixteen (i.e. 2^4) possible bit combinations and only ten of such are used to represent the decimal digits of 0 to 9.
- The 4-bit BCD code is used only to represent numbers.
- There are other variations of this coding system which make the representation of other characters possible. Examples are 6-bit, 7-bit and 8-bit codes

- The 6-bit BCD code allows for sixty-four bit combinations (i.e. 2^6).
- It could be used to represent the decimal digits 0 to 9,
- where we have the upper case alphabets A to F and twenty-eight special characters occurring.
- The four right-most bit positions are called numeric bit while the two left- most bit positions are the **ZONE bits**.
- The Zone bits are used in various combinations with numeric bits to represent numbers, letters and special characters.
- The eight-bit code allows 256 bit combinations (i.e. 2^8).

- This is known as the **Extended Binary Coded Interchange Code** (**EBCDIC**).
- This eight-bit EBCDIC can be used to represent decimal digits 0 to 9 having the upper case and lower case as letters and additional characters.
- In this code, the four left-most bit positions are the zone bits while the four right-most bit positions are the **NUMERIC** bits to represent numbers, letters and special characters.

- The **American Standard Code for Information Interchange Code (ASCII)** is a seven-bit code that was jointly developed by many computer manufacturer.
- There is an eight-bit version of the ASCII known as the ASCII known as the ASCII-8.
- Also very similar to EBCDIC and meets the requirements of computers designed to accept 8-bits.
- An n-bit number is binary number consisting of ‘n’ bits.
- Any decimal number in the range 0 to 2^{n-1} can be represented in binary form as an n-bit number.
- Every computer stores numbers, letters, and other special characters in binary form.
- There are several occasions when computer professionals need to know the raw data contained in a computer’s memory.
- A commonly used way to doing this is to print memory contents on a printer. This printout is called a memory dump.
- Memory dumps, which are in binary numbers, would have many pages of 0s and 1s.
- Working with these numbers would be very difficult and error prone for computer professionals.
- Hence, two number systems – octal and hexadecimal, are often used as shortcut notations for binary.

Octal Number System

- In octal number system, the base is 8.
- Hence, there are only eight symbols or digits:
- 0,1,2,3,4,5,6, and 7 (8 and 9 do not exist in this system).
- The largest single digit is 7 (one less than the base 8).
- Each position in an octal number represents a power of the base (8).
- Therefore, decimal equivalent of octal number 2057 (written as 2057) is: 8

- $(2 \text{ x } 8^3) + (0 \text{ x } 8^3) + (5 \text{ x } 8^1) + (7 \text{ x } 8^0) = 1024 + 0 + 40 + 7 = 1071$
- Since there are only 8 digits in octal number system,
- 3 bits will be (2^3= 8) which is sufficient to represent any number in a binary

Hexadecimal Number System

- In hexadecimal number system, the base is 16.
- Hence, there are 16 symbols or digits.
- The first 10 digits are the same digits of decimal number system which is 0, 1, 2,3,4,5,6,7,8, and 9.
- Assuming the remaining six digits are denoted by the symbols A, B, C, D, E, and F,
- Also representing decimal values 10,11,12,13,14 and 15, respectively.
- where the largest single digit is F or 15 (one less than the base 16).
- Hence each position in a hexadecimal number system will represents a power of the base (16). With this axioms we have therefore that
- Decimal equivalent of a hexadecimal number 1 AF (written as $1AF_{16}$) will be
- $1 \times 16^2) + (A \times 16^1) + (F \times 16^0) = (1 \times 256) + (10 \times 16) + (15 \times 1) = 256 + 160 + 15 = 431$
- Hence, $1AF_{16} = 431_{10}$
- Since there are only 16 digits in hexadecimal number 4 bits will be ($2^4 = 16$) which is sufficient to represent any hexadecimal number in a binary.

Conversion of Numbers

- Numbers expressed in decimal number system are much more meaningful to us, than are numbers expressed in any other number system.
- This is because we have been using decimal numbers in our day-to-day life, right from childhood.
- However, we can represent any number system in any other number system.
- since the input and final output values are to be in decimal and computer professionals are often required to convert numbers in other number system to decimal and vice-versa.
- Many methods can be used to convert numbers from one base to another.
- A method of converting from another base to decimal, and a method of converting from decimal to another base are described as follows:-
-

Converting from another Base to Decimal

- The following steps are sued to convert a number in any other base to a base 10

  (decimal) number.
- **Step 1 :** Determine the column (positional) value of each digit (this depends on the position of the digit and the base of the number system).
- **Step 2 :** Multiply the obtained column values (in Step 1) by the digits in the corresponding columns.
- **Step 3 :** Sum up the products calculated in Step 2. The total is the equivalent value in decimal.
- Example 1

  $1\ AC_{\mathbf{16}} = \mathbf{?}_{\mathbf{10}}$

  Solution :

  $1AC_{16} = 1 \times 16^2 + A \times 16^1 + C \times 16^0$ (since A is 10 and C is 12 from the Hexadecimal number), we then have

  $= 1 \times 256 + 10 \times 16 + 12 \times 1$

  $= 256 + 160 + 12 = 428_{10}$

Example 2 .

- $4052_{\mathbf{7}} = \mathbf{?}_{\mathbf{10}}$

  Solution :

  $4052_7 = 4 \times 7^3 + 0 \times 7^2 + 5 \times 7^1 + 2 \times 7^0$

  $= 4 \times 343 + 0 \times 49 + 5 \times 7 + 2 \times 1$

  $1372 + 0 + 35 + 2 = 1409_{10}$
-

Converting from Decimal to Another Base (Division- Remainder Technique)

- The Following steps are used to convert a have 10 (decimal) number to a number in another base.
- **Step 1 :** Divide the decimal number by the value of the new base. **Step 2 :** Record the remainder from Step 1 as the rightmost digit (least significant digit) of the new base number.
- **Step 3 :** Divide the quotient of the previous division by the new base.
- **Step 4:** Record the remainder from Step 3 as the next digit (to the left) of the new number. Repeat Steps 3 and 4, recording remainders from right to left, until the quotient becomes zero in Step 3.
- Note that the last remainder, thus obtained, will be the most significant digit of the new base number.

- For example

1. Convert 13 to base five
2. Convert 13 to 2
3. Convert 39 to 4
4. Convert 213 to base 10

- Solution

- Also note that it is possible to perform all the arithmetic operations such as addition, subtraction, multiplication and division in any base.
- Remember that the base in which you are working with, if you have to carry over or borrow any number, it is the multiples of the base you will carry or borrow.
- The base in which you are working should be indicated as the subscript of the number.
- For example

(a) $23_5 + 23_5 = 111_5$

(b) $26_9 - 18_9 = 7_9$

(c) $111_2 \times 11_2 = 10101_2$