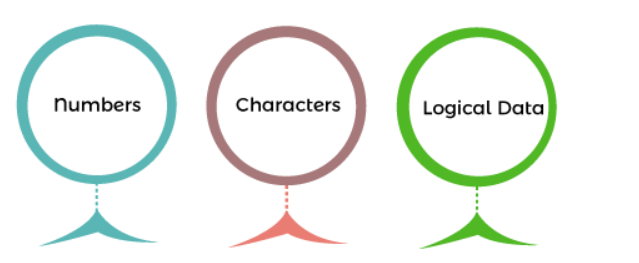
**Types of Operands**



Lecture 7

MSc. Riyam K. Marjan

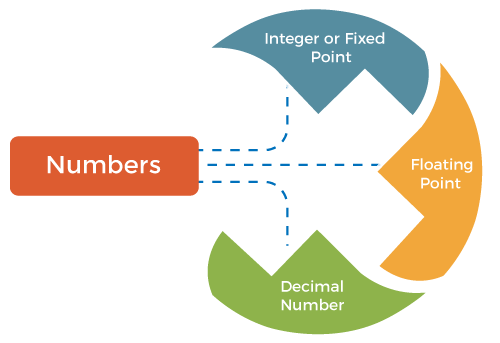
2022-2023

***Operands*** are definite elements of computer instruction that show what information is to be operated on. The most important general categories of data are

1. Numbers
2. Characters
3. Logical data

In many cases, some calculation must be performed on the operand reference to determine the main or virtual memory address.

### **Numbers**

All machine languages include numeric data types. Even in non-numeric data processing, numbers are needed to act as counters, field widths, etc. An important difference between numbers used in ordinary mathematics and numbers stored in a computer is that the latter is limited. Thus, the programmer is faced with understanding the consequences of rounding, overflow and underflow.

Here are the three types of numerical data in computers, such as:

1. **Integer or fixed point:** Fixed point representation is used to store integers, the positive and negative whole numbers (… -3, -2, -1, 0, 1, 2, 3, …).
2. **Floating point:** A Floating Point number usually has a decimal point, which means **0, 3.14, 6.5,** and**-125.5** are Floating Point
3. **Decimal number:** The decimals are an extension of our number system. We also know that decimals can be considered fractions with 10, 100, 1000, etc. The numbers expressed in the decimal form are called decimal numbersor decimals. For example:1, 4.09, 13.83, etc.

### **Characters**

A common form of data is text or character strings. While textual data are most convenient for humans. But computers work in binary. So, all characters, whether letters, punctuation or digits, are stored as binary numbers. All of the characters that a computer can use are called **character set**s. Here are the two common standards, such as:

1. American Standard Code for Information Interchange (ASCII)
2. Unicode

ASCII uses seven bits, giving a character set of 128 characters. The characters are represented in a table called the ASCII table. The 128 characters include:

* 32 control codes (mainly to do with printing)
* 32 punctuation codes, symbols, and space
* 26 upper-case letters
* 26 lower-case letters
* numeric digits 0-9

We can say that the letter 'A' is the first letter of the alphabet; 'B' is the second, and so on, all the way up to 'Z', which is the 26th letter. In ASCII, each character has its own assigned number. Denary, binary and hexadecimal representations of ASCII characters are shown in the below table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Character** | **Denary** | **Binary** | **Hexadecimal** |
| A | 65 | 1000001 | 41 |
| Z | 90 | 1011010 | 5A |
| a | 97 | 1100001 | 61 |
| z | 122 | 1111010 | 7A |
| 0 | 48 | 0110000 | 30 |
| 9 | 57 | 0111001 | 39 |
| Space | 32 | 0100000 | 20 |
| ! | 33 | 0100001 | 21 |

A is represented by the denary number 65 (binary 1000001, hexadecimal 41), B by 66 (binary 1000010, hexadecimal 42) and so on up to Z, which is represented by the denary number 90 (binary 1011010, hexadecimal 5A).

### **Logical data**

Normally, each word or other addressable unit (byte, half-word, and so on) is treated as a single unit of data. Sometimes, it is useful to consider an n-bit unit consisting of 1-bit items of data, each item having the value 0 or 1. When data are viewed this way, they are considered to be logical data.

The **Boolean** data can only represent two values: true or false. Although only two values are possible, they are rarely implemented as a single binary digit for efficiency reasons. Many programming languages do not have an explicit Boolean type, instead of interpreting 0 as false and other values as true. Boolean data refers to the logical structure of how the language is interpreted to the machine language. In this case, a Boolean 0 refers to the logic False, and true is always a non zero, especially one known as Boolean 1.

AD