CHAPTER - 1
INTRODUCTION TO C PROGRAMMING

CHAPTER 1 *(DAY 1 Fore noon)*

INTRODUCTION TO C PROGRAMMING

C LANGUAGE	(Reading)
STRUCTURED PROGRAMMING	(Reading)
DATA TYPE	
DATA STRUCTURES AND C	
C ENVIRONMENT	(Reading)
ALGORITHM	(Reading)
PSEUDOCODE	(Reading)
C PROGRAM STRUCTURE	(Reading)
FIRST C PROGRAM	
INDENTATION	(Reading)
USING COMMENTS	(Reading)

INTRODUCTION TO C PROGRAMMING

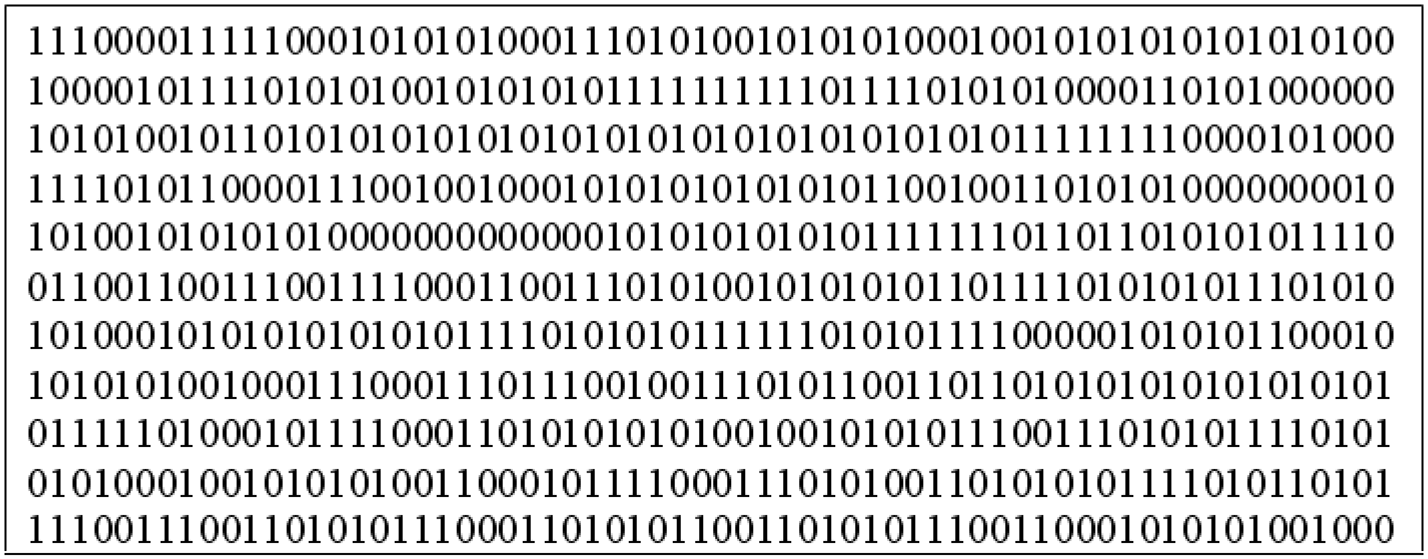
DATA TYPES

- **In general, for many high level languages, a *data type* is the property or attributes of data.**
- **Each *data type* will have some values associated and some operations those can be performed on the data values.**
- **Before using a data in the program, the data variable has to be defined first with an appropriate *data type*.**
- **C language is very specific in type checking.**
- **The operations on the data also define what type of problems can be solved with a given *data type*.**

INTRODUCTION TO C PROGRAMMING

DATA TYPES

A sample partial memory in a computer:



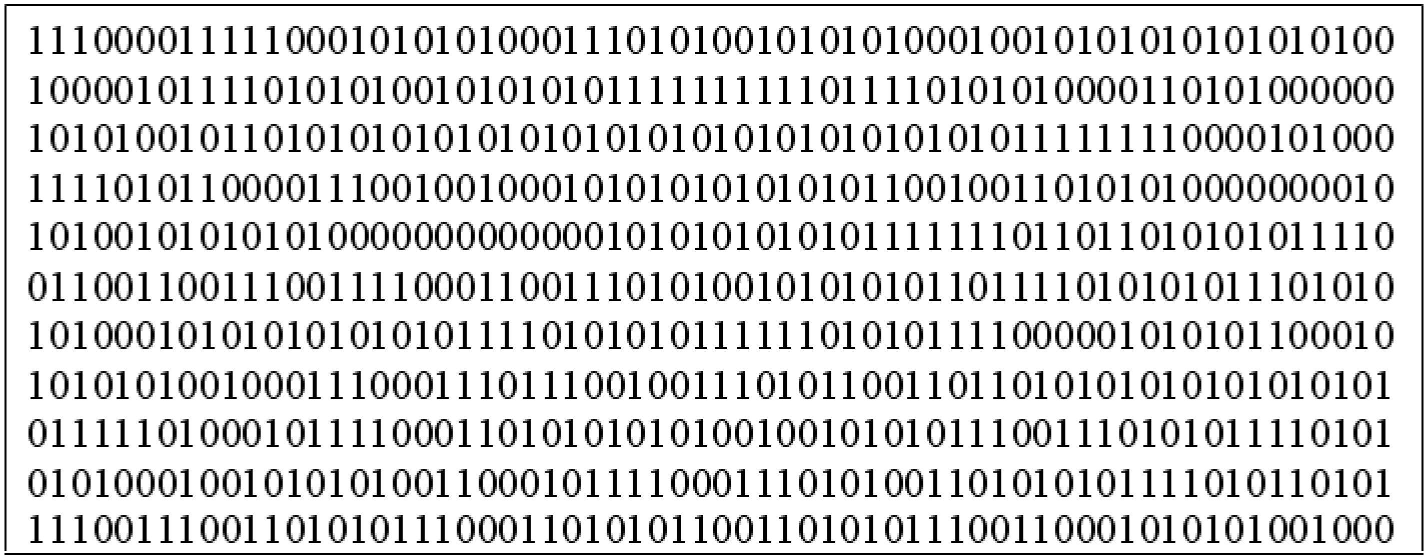
```
11100001111100010101010001110101001010101000100101010101010100
10000101111010101001010101011111111110111101010100001101010000000
10101001011010101010101010101010101010101010111111110000101000
11110101100001110010010001010101010101100100110101010000000010
10100101010101000000000000001010101010101111111011011010101011110
01100110011100111100011001110101001010101011011110101010111101010
1010001010101010101011110101010111111010101111000001010101100010
10101010010001110001110111001001110101100110110101010101010101
01111101000101111000110101010101001001010101110011101010111110101
0101000100101010100110001011110001110101001101010101111010110101
1110011100110101011100011010101100110101011100110001010101001000
```

- The sample partial memory of a computer is shown in the course material, which shows the values in ‘1’ and ‘0’.
- The program interprets the bit pattern in the memory depending on the data type it is dealing with.
- Data types will be revisited when we discuss about abstract data types.

INTRODUCTION TO C PROGRAMMING

DATA TYPES

A sample partial memory in a computer:



```
1110000111110001010101000111010100101010100010010101010101010100
1000010111101010100101010101111111111011110101010000110101000000
1010100101101010101010101010101010101010101010101111111110000101000
1111010110000111001001000101010101010101100100110101010000000010
1010010101010100000000000000101010101010111111101101101010101011110
0110011001110011110001100111010100101010101101111010101011101010
1010001010101010101011110101010111111010101011110000010101011000010
1010101001000111000111011100100111010110011011010101010101010101
0111110100010111100011010101010100100101010111001110101011110101
0101000100101010100110001011110001110101001101010101011110101101
1110011100110101011100011010101100110101011100110001010101001000
```

- The sample partial memory of a computer is shown in the course material, which shows the values in ‘1’ and ‘0’.
- The program interprets the bit pattern in the memory depending on the data type it is dealing with.
- Data types will be revisited when we discuss about abstract data types.

INTRODUCTION TO C PROGRAMMING

DATA STRUCTURES AND C

- **The study of *data structures* involves identifying and developing mathematical entities and using these entities and operations including determining what classes of problems can be solved.**
- **It also involves determination of representations for those abstract entities and to implement the abstract operations on these concrete representations.**
- ***Building data structures* involves developing higher level of data structures from already known data structures.**
- **Some of the already known data structures would be the primitive *data types* supported by the language itself.**
- **Using efficient algorithms and newly developed data structures enhance the effectiveness of programs to solve software problems.**
- **Definition of formal data types with their values and operations need to be performed on those values would help understand the behavior of newly evolved data structures.**

INTRODUCTION TO C PROGRAMMING

FIRST C PROGRAM

```
#include<stdio.h>

int main()
{
    printf ("Hello World\n");
    return 0;
}
```

- #include is a preprocessor directive.
- stdio.h lets user call system commands to read input to the program and write output from the program.
- *main* is a starting function of a program.

INTRODUCTION TO C PROGRAMMING

FIRST C PROGRAM

In C language all *system calls* are functions.

- Every function has to return something to the calling program including main.
- Functions may or may not take arguments depending on how they are defined.
- {} Curly braces are used for multiple statements under any statement.
- *printf* is a system call for writing output formatted.
- *return* statement will return whatever is defined at the function definition. In this case main function is defined to return an integer value.