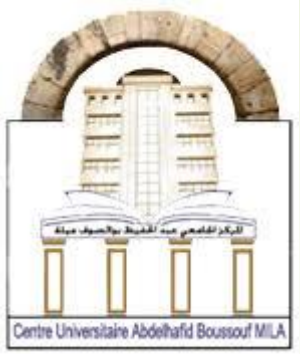
Abdelhafid Boussouf
University
Center of Mila



Structure of Computers and Applications
1st year ST – ENGINEERING

Part 2: The basics of Algorithm and Program

Course 06: C Language

By

Dr. Farouk KECITA

Academic year : 2024/2025

7- Introduction to C Language

What is a program/ programming language ?

- A **program** is a sequence of instructions that a computer follows to solve a problem
- A **programming language** is a set of words and symbols and codes that enables human to write a computer program.

What is C?

- C is a programming language developed at **AT & T‟s Bell Laboratories of USA in 1972**. It was designed and written by **Dennis Ritche**. **Dennis Ritchie** is known as the **founder of c language**.

Features of C

1. Portability or machine independent
2. Sound and versatile language
3. Fast program execution.
4. An extendible language.
5. Tends to be a structured language.

8- STRUCTURE OF A C PROGRAM

Header	#include <stdio.h>
main()	int main() {
Variable declaration	int a = 10;
Body	printf("%d ", a);
Return	return 0; }

8.1 **Header Files Inclusion**: The first and foremost component is the inclusion of the Header files in a C program.

- A header file is a file with **extension .h** which contains C function declarations and macro definitions to be shared between several source files.
- Some of C Header files:

- **stdio.h** – Defines core input and output function.
- **stdlib.h** – Defines numeric conversion functions.
- **math.h** – Defines common mathematical functions.

- **Syntax to include a header file in C:** #include

8- STRUCTURE OF A C PROGRAM

8.2 Main Method Declaration: The next part of a C program is to declare the main() function.

Syntax to Declare main method:

```
int main()
{   }
```

8.3 Variable Declaration: It refers to the variables that are to be used in the function, the variables are to be declared before any operation in the function.

Example:

```
int main() {
    int a;
    . . .
```

8.4 Body: Body of a function in C program, refers to the operations that are performed in the functions. It can be anything like manipulations, searching, sorting, printing, etc.

Example:

```
int main() {
    i nt A;
    printf("%d", A);
    . .
```

8- STRUCTURE OF A C PROGRAM

8.5 Return Statement: The return statement refers to the returning of the values from a function. This return statement and return value depend upon the return type of the function.

- For example, if the return type is void, then there will be no return statement. In any other case, there will be a return statement and the return value will be of the type of the specified return type.

Example:

```
int main() {
    int A;
    printf("%d", A);
    return 0; }
```

- **Writing first program:**

```
#include <stdio.h>
int main()
{
int number1, number2, sum;
printf("Enter two integers: ");
scanf("%d %d", &number1, &number2)
sum = number1 + number2;
printf("%d + %d = %d", number1, number2, sum);
return 0;
}
```

9- KEYWORDS IN C

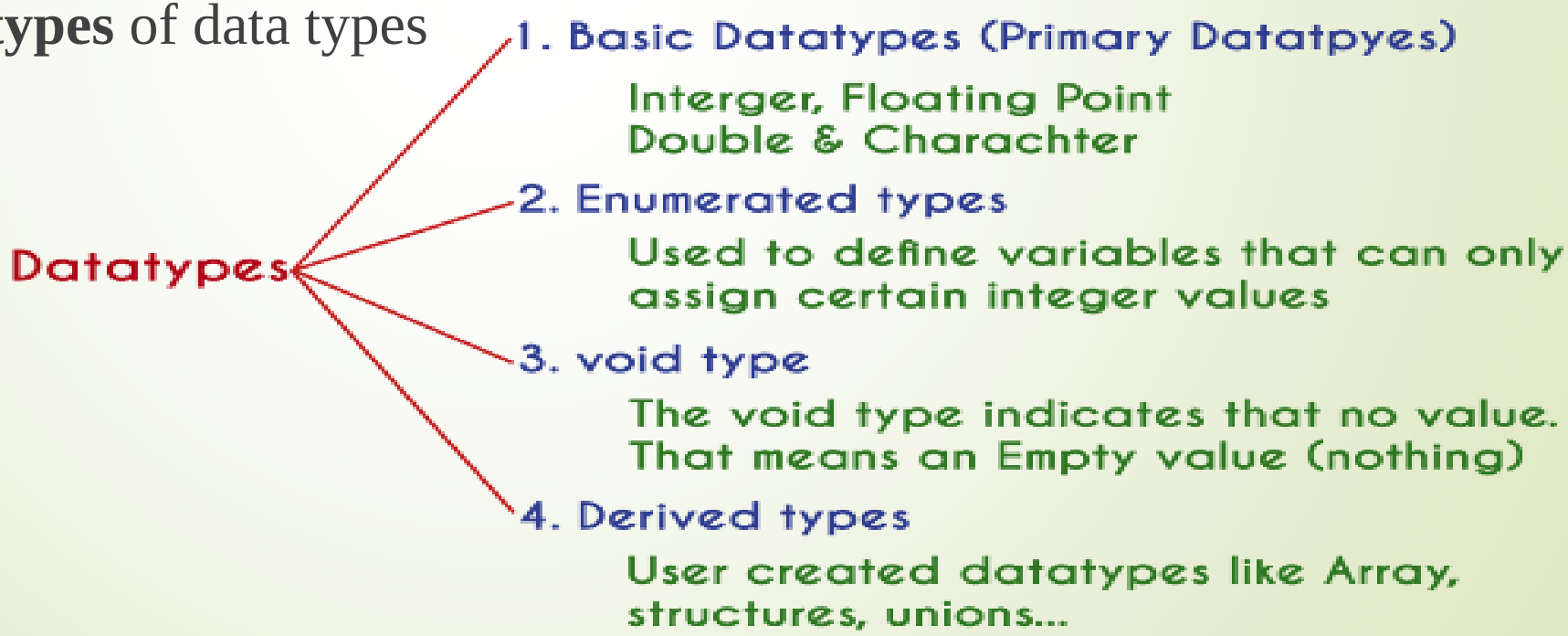
- A keyword is a **reserved word.**
- All keywords have fixed meaning that means we cannot change.
- Keywords serve as basic building blocks for program statements.
- All keywords must be written in lowercase.
- A list of 32 keywords in c language is given below:

double- float- int- short- struct- unsigned-break- continue -else forlong- signed- switch- void-case- default- enum – goto - registersizeof- typedef- volatile- char- do- extern- if- return- static –unionwhile- Auto- const.

10- DATA TYPES / TYPES IN C

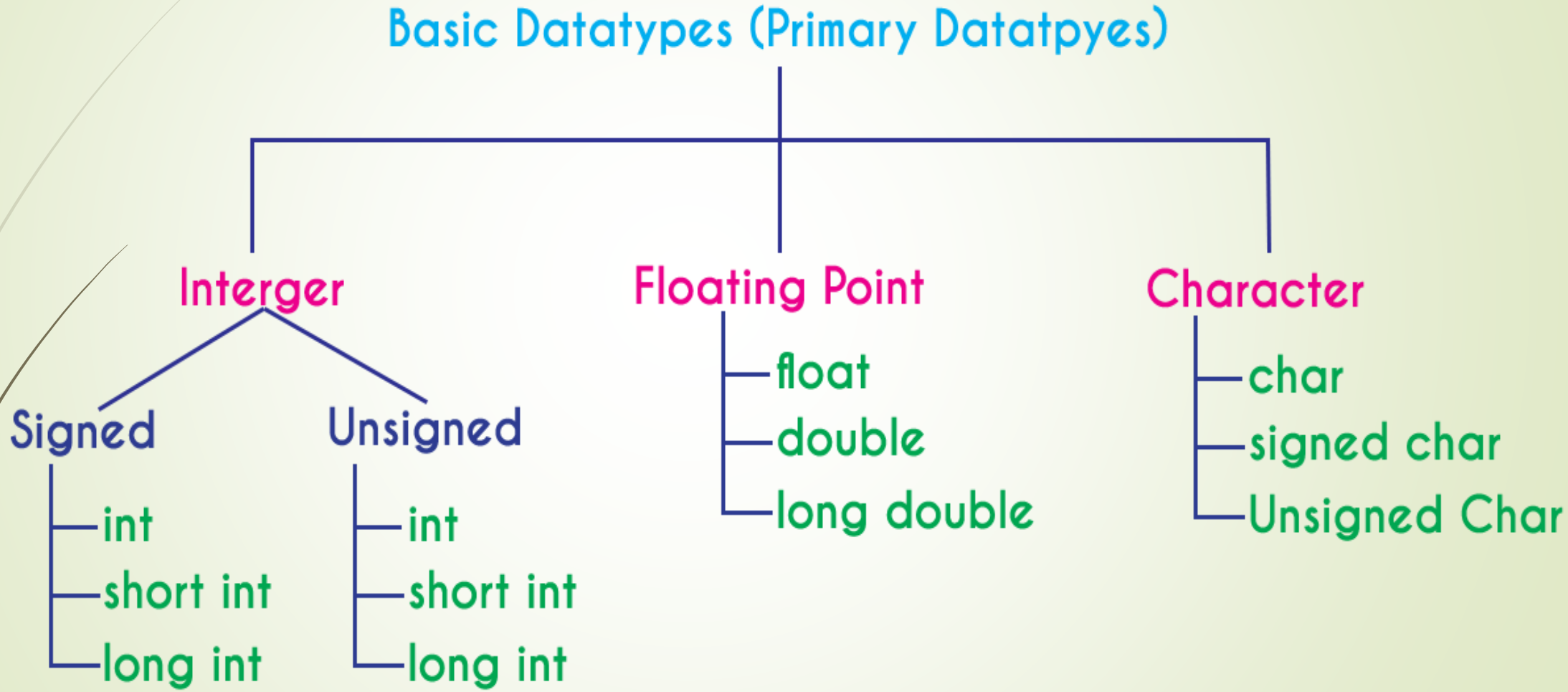
- What are the data types in C?

- Data types refer to an extensive system used for **declaring/defining** variables or functions of different types before its use.
- The type of a variable determines how much **space** it occupies in storage and how the **bit** pattern stored is interpreted.
- There are **4 types** of data types



10- DATA TYPES / TYPES IN C

❑ Basic Data Types



11- VARIABLES

- A variable is a name of memory location. It is used to store data. Variables are changeable, we can change value of a variable during execution of a program.
- It can be reused many times.

Note: Variable are nothing but identifiers.

Rules to write variable names:

1. A variable name contains maximum of 30 characters/ Variable name must be upto 8 characters.
2. A variable name includes alphabets and numbers, but it must start with an alphabet.
3. It cannot accept any special characters, blank spaces except under score(_).
4. It should not be a reserved word.

Example:

a num1 MAX min St_name StName class_mark

11- VARIABLES

- **Declaration of Variables :** A variable can be used to store a value of any data type. The declaration of variables must be done before they are used in the program.

- The general format for declaring a variable.
  Syntax : data_type variable-1,variable-2,------, variable-n;

- Variables are separated by **commas** and declaration statement ends with a **semicolon**.
  Example: int x,y,z; float a,b; char m,n;

- **Assigning values to variables :** values can be assigned to variables using the assignment operator (=).

- The general format statement is :
  Syntax : variable = constant; OR data_type variable = constant;

- **Example** : x=10; a= 10.2; n="k"; OR int x=10; float a= 10.2; char n="k";

12- CONSTANTS

- Constants **refer** to **fixed** values that do **not change** during the execution of a program.
  Note: constants are also called literals**.**

❑ declaring constant:

There are two ways of declaring constant:

✓ Using **const** keyword

Syntax to Define Constant: **const data_type var_name = value;**

Example: const int int_const = 25;

const float float_const = 15.66;

✓ Using **#define** pre-processor

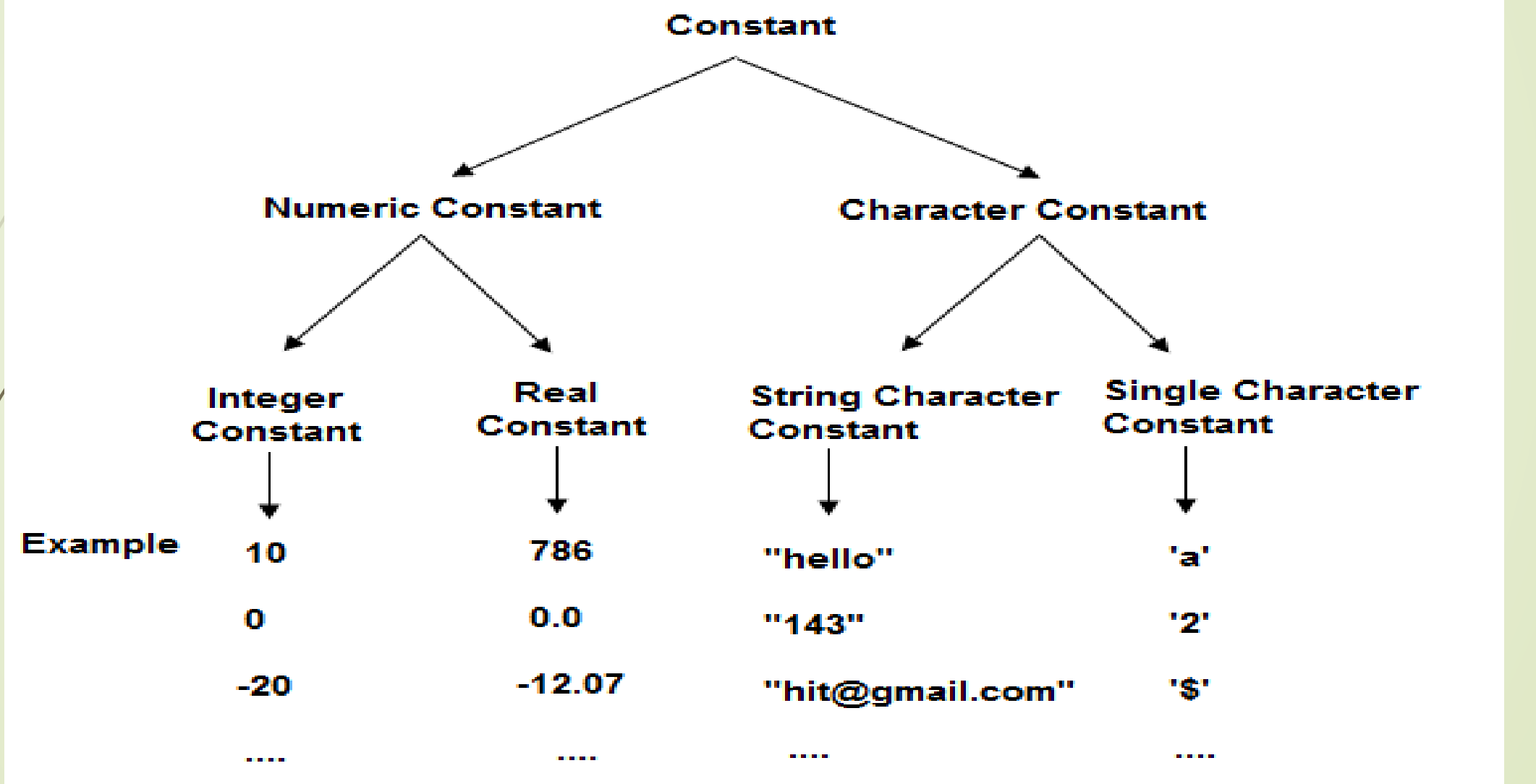
Syntax : #define const_name value

Example:

```
#include <stdio.h>
#define pi 3.14
int main() {
 printf("The value of pi:%.2f",pi);
 return 0; }
```

12- CONSTANTS

❑ Types of Constants in C



Special characters and their purpose

Backslash Character	Meaning
\n	New line
\t	Horizontal tab
\v	Vertical tab
\0	Nul value

Data Types conversion Specifiers:

Specifier	Meaning
%d (or) %i	Used to read and displays integer values
%f	Used to read and display real values in decimal notations. Example: 10.256
%e	Used to read and display real values in scientific notation. Example: 3.6e7
%E	Used to read and display real values in scientific notation with capital letter. Example: 3.6E7
%c	Used to read and display characters
%lf	Used to read and display long double values