SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECB211 – Microcontroller Programming & Interfacing

II YEAR/ IV SEMESTER

1

UNIT 3 – PIC PROGRAMMING IN C

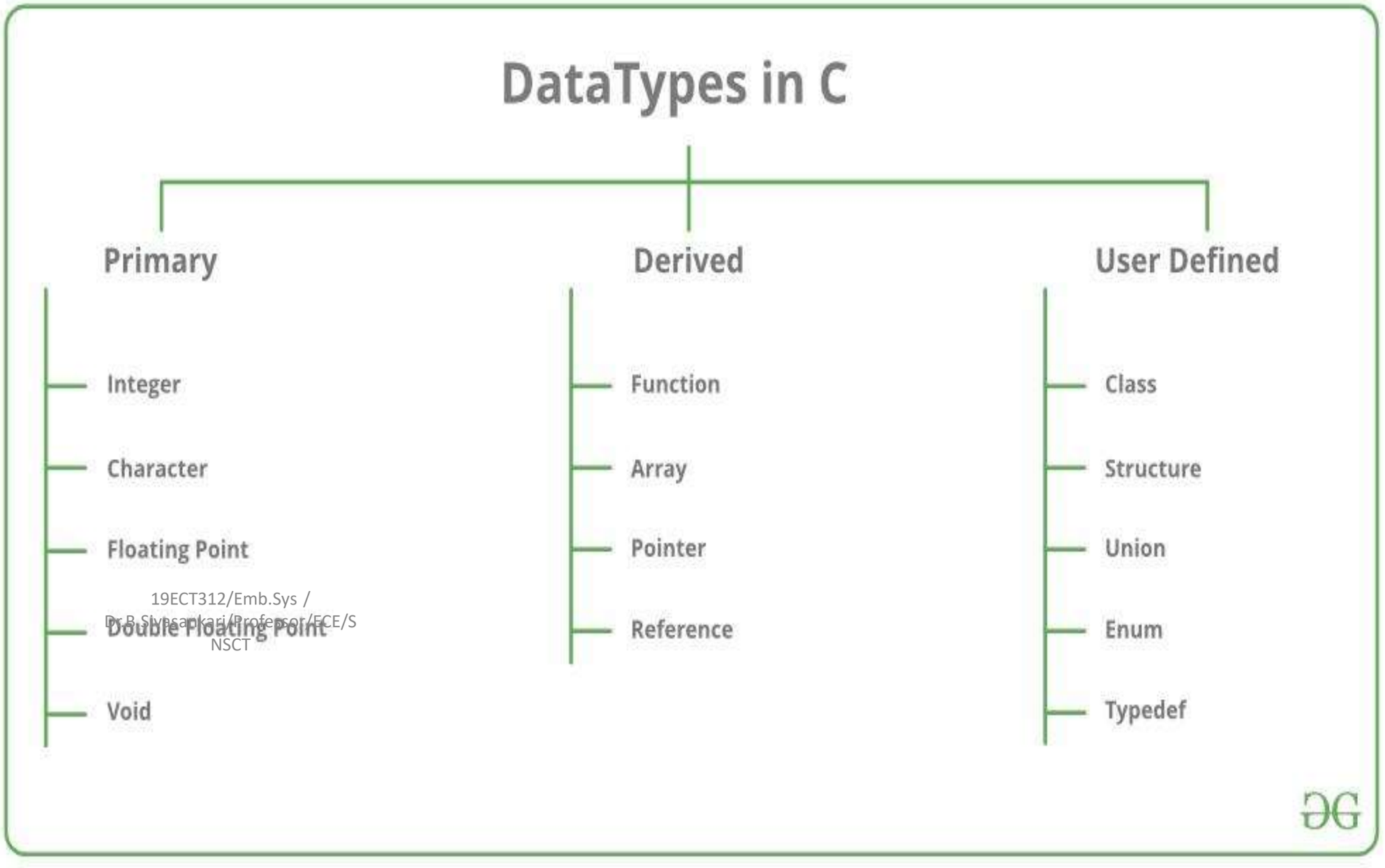
TOPIC 1 – Data Types and Time Delays in C

DATA TYPES IN C

- The data type in C defines the amount of storage allocated to variables ,the values that they can accept ,and the operation that can be performed on those variables.
- C is rich in data types.
- The verity of data type allow the programmer to select appropriate data type to satisfy the need of application as well as the needs of different machine.
- Three classes of Data-Type

1. Primary Data Type
2. Derived Data Type
3. User Defined Data Type

Primary Data Types
(Fundamental Data Types)



Size and Range of Data-Types on a 16-bit machine

Integer (value) - One word (space)

Word size will vary between 16 bits and 32 bits machines

16 bit machine : Int 2 bytes(16 bits)

16 bit machine : Int 4 bytes(32 bits)

Ranges of Integers : 65538 (2 power 16) – 0 to 65537

TYPE	SIZE	RANGE
(Signed)short int (% hd)	1(8 bits)	128 to 127
unSigned short int (% hu)	1(8 bits)	0 to 255
Int(or) signed int (% d)	2(16 bits)	32,768 to 32,767
Unsigned int (% u)	2(16 bits)	0 to 65537
Signed long int (% ld)	4 (32 bits)	2,147,483,648 to 2,147,483,647
Unsigned long int (% lu)	4 (32 bits)	0 to 4,294,967,265

Character and Floating Point Data Types

Character

- Just like we use a set of various words, numbers, statements, etc., in any language for communication, the C programming language also consists of a set of various different types of characters.
- These are known as the characters in C. They include digits, alphabets, special symbols, etc.
- The C language provides support for about 256 characters - **% C**

Signed cha 1 byte (8 bits) - 128 to 127

Unsigned cha 1 byte (8 bits) 0 to 255

ASCII – American Standard Code for Information Interchange

Character and Floating Point Data Types

Types of Characters in C

- The C programming language provides support for the following types of characters

In other words, these are the *valid* characters that we can use in the C language

Digits
Alphabets
Main Characters

All of these serve a different set of purposes, and we use them in different contexts in the C language.

Character and Floating Point Data Types

Alphabets

- The C programming language provides support for all the alphabets that we use in the English language.
- C program would easily support a total of 52 different characters- 26 uppercase and 26 lowercase

Type of Character	Description	Characters
Lowercase Alphabets	a to z	a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
Uppercase Alphabets	A to Z	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z

Character and Floating Point Data Types

Digits

- The C programming language provides the support for all the digits that help in constructing/ supporting the numeric values or expressions in a program.
- These range from 0 to 9, and also help in defining an identifier.
- Thus, the C language supports a total of 10 digits for constructing the numeric values or expressions in any program.

Type of Character	Description	Characters
Digits	0 to 9	0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Character and Floating Point Data Types

Special Characters

➢We use some special characters in the C language for some special purposes, such as logical operations, mathematical operations, checking of conditions, backspaces, white spaces, etc.

➢We can also use these characters for defining the identifiers in a much better way. For instance, we use underscores for constructing a longer name for a variable, etc.

➢The C programming language provides support for the following types of special characters

Type of Character	Examples
Special Characters	` ~ @ ! $ # ^ * % & ( ) [ ] { } < > + = _ – | / \ ; : ‘ “ , . ?

Summary of Special Characters in C

Type of Character	Description	Characters
Lowercase Alphabets	a to z	a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
Uppercase Alphabets	A to Z	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z
Digits	0 to 9	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Special Characters	–	` ~ @ ! $ # ^ * % & ( ) [ ] { } < > + = _ – | / \ ; : ‘ “ , . ?
White Spaces	–	Blank Spaces, Carriage Return, Tab, New Lin

Floating Point Data Types - FLOAT

- In C, float data type occupies 4 bytes (32 bits) of memory to store real numbers that have at least one digit after the decimal point
- A float data type can hold any value between 3.4E-38 to 3.4E+38
- The floating-point variable has a precision of 6 digits i.e., it uses 6 digits after the decimal point

```c
#include <stdio.h>
void main() {
    float a = 2.135;
    float b = 7.217;
    float sum;
    sum = a + b;
    printf("Sum of two numbers = %f\n", sum);
}
```

Floating Point Data Types - FLOAT

- 4 bytes of memory is allocated to each variable a, b and they are initialized with floating-point constants 2.135000 and 7.217000 respectively
- A floating-point variable can represent a wider range of numbers than a fixed-point variable of the same bit width at the cost of precision
- A signed 32-bit integer variable has a maximum value of $2^{31} - 1 = 2,147,483,647$, whereas an IEEE 754 32-bit base-2 floating-point variable has a maximum value of $(2 - 2^{-23}) \times 2^{127} \approx 3.4028235 \times 10^{38}$
- All integers with 7 or fewer decimal digits, and any 2^n for a whole number $-149 \leq n \leq 127$, can be converted exactly into an IEEE 754 single-precision floating-point value
- There's no hardware support for unsigned floating-point operations. So, C doesn't offer it.

Floating Point Data Types - Double Data type

- In C, a double data type is used to increase the accuracy of the real number wherever a float is not sufficient.
- A double data type occupies 8 bytes (64 bits) of memory to store real numbers, which have at least one digit after the decimal point.
- A double data type can hold any value between 1.7E-308 to 1.7E+308.Double data type values have a precision of 14 digits i.e., they can have 14 digits after the decimal point. Consider the following example using a double data type

```c
#include <stdio.h>
void main() {
    double num1 = 26.7368;
    double num2 = 1.42924;
    double sum;
    sum = num1 + num2;
    printf("Sum of the numbers = %f\n", sum);
}
```

Floating Point Data Types - Double Data type

- 8 bytes of memory is allocated to each variable num1, num2 and they are initialized with real number constants 26.7368 and 1.42924 respectively

- To further extend the precision of a double data type, the user can use a long double data type
- The long double type is guaranteed to have more bits than a double, while the exact number may vary from one hardware platform to another

- A long double data type allocates 10 bytes (80 bits) of memory to store the given values. A long double data type can hold any value between 3.4E-4932 to 1.1E+4932

- In a 32 – bit forth implementation,doubke data type has a range of -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 (signed) or 0 to 18,446,744,073,709,551,615 (unsigned)

Derived Data Type

- The derived data types are basically derived out of the fundamental data types. A derived data type won't typically create a new data type – but would add various new functionalities to the existing ones instead

Uses of Derived Data types in C

- We can derive the derived data types out of the primitive data type by adding some extra relationships to the elements that are available with the primitive data types.

- We use the derived data types to represent multiple values as well as single values in a program

Types of Derived Data Types in C

The C language supports a few derived data types

Arrays – The *array* basically refers to a sequence (ordered sequence) of a finite number of data items from the same data type sharing one common name.

Function – A *Function in C* refers to a self-contained block of single or multiple statements. It has its own specified name.

Pointers – The *Pointers in C* language refer to some special form of variables that one can use for holding other variables' addresses.

Unions – The unions are very similar to the structures. But here, the memory that we allocate to the largest data type gets reused for all the other types present in the group

Structures – A collection of various different types of data type items that get stored in a contagious type of memory allocation is known as *structure in C*

Types of Derived Data Types in C - Array

- An array is a group of similar elements or data items of the same type collected at contiguous memory locations. In simple words, we can say that in computer programming, arrays are generally used to organize the same type of data

Array for Integral value:

Array for Character value:

Types of Derived Data Types in C - Array

Types of Arrays

One-Dimensional Arrays
A one-dimensional array is a kind of linear array. It involves single sub-scripting.

Multi-Dimensional Arrays

Two-Dimensional Arrays:An array involving two subscripts [] [] is known as a two-dimensional array. They are also known as the array of the array. Two-dimensional arrays are divided into rows and columns and are able to handle the data of the table .

Three-Dimensional Arrays:When we require to create two or more tables of the elements to declare the array elements, then in such a situation we use three-dimensional arrays.

Types of Derived Data Types in C - Array

Advantages of Array

- It is a better version of storing the data of the same size and same type.
- It enables us to collect the number of elements in it.
- Arrays have a safer cache positioning that improves performance.
- Arrays can represent multiple data items of the same type using a single name.

Disadvantages Of Array:

- In an array, it is essential to identify the number of elements to be stored.
- It is a static structure. It means that in an array, the memory size is fixed.
- When it comes to insertion and deletion, it is a bit difficult because the elements are stored sequentially and the shifting operation is expensive.

Types of Derived Data Types in C - Function

Use of the Function in C

One can easily divide a C program code into several separate functions. How we divide our available code according to different available functions is totally up to us. But the division should occur logically in such a way that every function is capable of performing a specific task.

The declaration of function informs the compiler about the name of the function, parameters, and return type. In other words, the function definition helps in providing the actual body of any function that we want in a program/ code.

We can also refer to functions as a sub-routine, a method, or a procedure in a program. The standard library in C language provides its users with various built-in functions that the concerned programs can call. For example, we can use the strcat() function for concentrating two strings, copy the location of one memory to another location of memory using the memcpy() function, and many more.

Advantages of Using Functions in C Programming

The functions in the C programming language offer the following advantages to us:

- When we make use of the functions, then we can easily avoid the rewriting of the same code/ logic time and again multiple times in any program.
- The calling of C functions may appear as many numbers of times as we want in any program. And we can do so from any place in the program that is given to us.
- We can perform the tracking of a large C program pretty easily if we divide it into various functions.
- One of the primary achievements of the C functions is reusability.
- However, remember that the function calling always acts as an overhead in the case of a C program.

Types of Derived Data Types in C - Function

Types of Functions

The C programming language has functions that are of the following types:

- **User-defined Functions –** These are the types of functions that we can create using the C programmer so that we can make use of it multiple times. This function reduces the complexity of any big program- and thus, optimizes the given code.
- **Library Functions –** These are the functions whose declaration occurs in the header files of C, such as floor(), ceil(), outs(), gets(), printf(), scanf(), etc.

Types of Derived Data Types in C - Pointers &Unions

- The pointers in C language refer to the variables that hold the addresses of different variables of similar data types.
- We use pointers to access the memory of the said variable and then manipulate their addresses in a program.
- The pointers are very distinctive features in C- it provides the language with flexibility and power.

Types Of Pointers

1. The Null Pointer : The null pointer has a value of 0

2. **The Void Pointer** :The void pointer is also known as the generic pointer in the C language. This pointer has no standard data type

3. **The Wild Pointer** :We can call a pointer a wild pointer if we haven't initialized it with anything. Such wild pointers are not very efficient in a program

Unions – The unions are very similar to the structures. But here, the memory that we allocate to the largest data type gets reused for all the other types present in the group.

User- Defined Data Types in C

- The UDT (User-Defined Data Type) is a typical data type that we can derive out of any existing data type in a program.
- We can utilise them for extending those built-in types that are already available in a program, and then you can create various customized data types of your own

Why do we Need User-Defined Data Types in C?

- The data types originally present in a program may not offer a wide variety of functions. But despite the various basic as well as derived data types present in the C language, there is a special feature using which we can define custom data types of our own, on the basis of our needs.

- The User-Defined Data Types are basically defined by a user according to their will. These offer various functions on the basis of how one defines them. Thus, these are termed to be “User-Defined”

User- Defined Data Types in C

The C program consists of the following types of UDT

Structures
Union
Typedef
Enum

Structures

We use the structure for organizing a group of various data items into one single entity – for grouping those data items that are related but might belong to different data types.

The structure data types are related (usually), such as the different sets of information regarding a person, an account, or a part, etc. Every data item present in a structure is known as a member. These members are sometimes also known as fields.

User- Defined Data Types in C

Union

A union is basically a collection of various different data types that are present in the C language, but not very similar to each other.

The union mainly allows the storage of those data types that are different from each other in the very same memory location.

Any user will be able to define a union using various members, but at any given time, just a single member would be able to contain the given value.

The unions provide us with an efficient way in which we can use the very same memory location for multiple purposes.

User- Defined Data Types in C

Typedef

We use the keyword typedef for creating an alias (a new name) for a data type that already exists. The typedef won't create any new form of data type.
When using the typedef data type, the syntax would be:
typedef existing_data_type new_type;

Enum

The enum refers to a keyword that we use for creating an enumerated data type. The enum is basically a special data type (enumerated data type) that consists of a set of various named values – known as members or elements. We mainly use it for assigning different names to the integral constants. This way, the program becomes much more readable.
Here is the format that we use for creating the enum type:
enum identifier (value_a, value_b, …. , value_z);

Time Delays in C

What is time delay in C programming?
Delay in C: delay function is used to suspend execution of a program for a particular time.
Declaration: void delay(unsigned int); Here unsigned int is the number of milliseconds (remember 1 second = 1000 milliseconds)

```c
#include<stdio.h>

int main()
{
    int c, d;
    for (c = 1; c <= 32767; c++)
        for (d = 1; d <= 32767; d++)
        {}

    return 0;
}
```

THANK YOU