



SNS COLLEGE OF TECHNOLOGY



Coimbatore-36.

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

COURSE CODE AND NAME : 23IT101 C Programming and Data structures

I YEAR/ I SEMESTER

UNIT – I INTRODUCTION TO C

Topic:DATA TYPE

Mrs. S.PRIYADARSINI

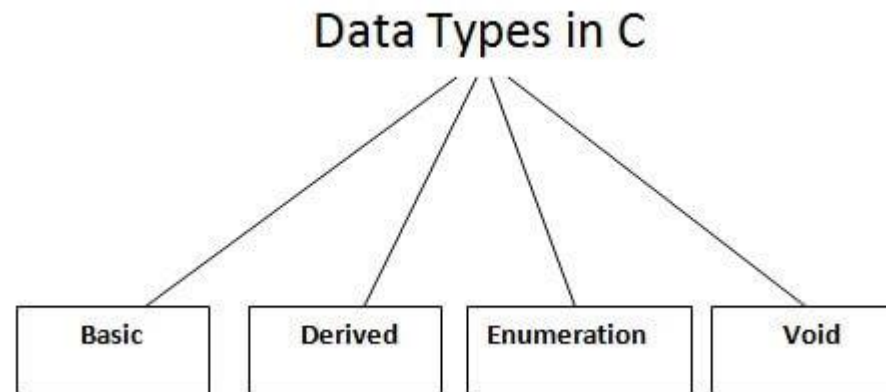
Assistant Professor

Department of Computer Science and Engineering



Data Type

- A data type specifies the type of data that a variable can store such as integer, floating, character, etc.
- Each variable in C has an associated data type.
- It specifies the type of data that the variable can store like integer, character, floating, double, etc.
- Each data type requires different amounts of memory and has some specific operations which can be performed over it.





Types of Data Type

- **Primitive Data Type**-Primitive data types are the most basic data types that are used for representing simple values such as integers, float, characters, etc. Eg:int, char, float, double, void
- **Derived Type**- The data types that are derived from the primitive or built-in data types are referred to as Derived Data type Eg: array, pointers, function
- **User Defined Data Type**-The user-defined data types are defined by the user himself. Eg:structure, union, enum



Primitive Data Type-*Integers*

- ***Integers*** are entire numbers without any fractional or decimal parts, and the ***int data type*** is used to represent them.
- It is frequently applied to variables that include ***values***, such as ***counts, indices***, or other numerical numbers. The ***int data type*** may represent both ***positive*** and ***negative numbers*** because it is signed by default.
- An ***int*** takes up ***4 bytes*** of memory on most devices, allowing it to store values between around -2 billion and +2 billion.



Char:

- Individual characters are represented by the ***char data type***. Typically used to hold ***ASCII*** or ***UTF-8 encoding scheme characters***, such as ***letters, numbers, symbols, or commas***. There are **256 characters** that can be represented by a single char, which takes up one byte of memory. Characters such as **'A', 'b', '5', or '\$'** are enclosed in single quotes.

Float:

- To represent integers, use the ***floating data type***. Floating numbers can be used to represent fractional units or numbers with decimal places.



- The ***float type*** is usually used for variables that require very good precision but may not be very precise. It can store values with an accuracy of about ***6 decimal places*** and a range of about ***3.4 x 10³⁸*** in ***4 bytes*** of memory.

Double:

- Use two data types to represent ***two floating integers***. When additional precision is needed, such as in scientific calculations or financial applications, it provides greater accuracy compared to float.
- ***Double type***, which uses ***8 bytes*** of memory and has an accuracy of about ***15 decimal places, yields larger values***. C treats floating point numbers as doubles by default if no explicit type is supplied.



```
int age = 25;
```

```
char grade = 'A';
```

```
float temperature = 98.6;
```

```
double pi = 3.14159265359;
```

Derived Data Type

- Beyond the fundamental data types, C also supports ***derived data types***, including ***arrays, pointers, structures, and unions***.
- These data types give programmers the ability to handle heterogeneous data, directly modify memory, and build complicated data structures.



Array

- An **array, a derived data type**, lets you store a sequence of **fixed-size elements** of the same type. It provides a mechanism for joining multiple targets of the same data under the same name.
- The index is used to access the elements of the array, with a **0 index** for the first entry. The size of the array is fixed at declaration time and cannot be changed during program execution. The array components are placed in adjacent memory regions.



- › Logical operators are used for testing more than one condition and making decisions.

OPERATOR	MEANING
&&	Logical AND
	Logical OR
!	Logical NOT



Logical Operator

```
#include<stdio.h>
void main()
```

```
int a, b;
printf("Enter values for a and b : ");
scanf("%d %d", &a, &b);
printf("\n %d", (a<b)&&(a!=b));
printf("\n %d", (a<b)|| (b<a));
printf("\n %d',!(a==b));
```



ASSIGNMENT OPERATORS

- > These operators are used for assigning the result of an expression to a variable.

- > b=a; OPERATORS:

==, +=, *=, /=, %=,



```
#include<<stdio.h>>
void main()

int a, b, c;
printf("Enter the values for a and b : ");
scanf("%d %d",&a,&b);
printf("\n the values of += is:%d",c=a+b);
printf("\n the values of +=is:%d",c+=b);
printf("\n the value of -= is:%d",c-=a);
printf("\n the value of *=is:%d",c*=a);
printf("\n the value of /=is:%d",c/=b);
printf("\n the value of %=is:%d",c%=b);
```



INCREMENT & DECREMENT OPERATORS

- Two most useful operators which are present in 'c'
- are increment and decrement operators.
- Operators: `*+` and `--`
- The operator `*+` adds one to the operand
- The operator `--` subtracts one from the operand.
 - Both are unary operators and can be used as pre or
 - post increment/decrement.



Special Operators Contd...

Sizeof Operator:

Sizeof is an operator used to return the number of bytes the operand occupies.

Syntax:

Int a;

Sizeof(a) -2

Float v;

Sizeof(v) -4



*Thank
you!*