C Programming

Trainer : Nisha Dingare

Email : nisha.dingare@sunbeaminfo.com

Tokens :

- A token is a smallest individual element in a program. Each and every punctuation and word that you come across in a C program is token.
    - Keywords
    - Data Types
    - Identifiers
    - Variables
    - Constants
    - Operators

Identifiers :

- Identifiers are simply the names that are used to identify the variables, functions, macros etc.

- There are certain rules to be followed for identifiers
  - It should start with the alphabet or underscore (_).
  - No commas, blank spaces or special symbols except (_) are allowed.
  - Identifiers are Case Sensitive.

- Examples:
  1. Var_1 (Valid)
  2. 1_var (Not Valid)
  3. _var (valid)
  4. Var-1 (invalid)
  5. Basic Salary (invalid)

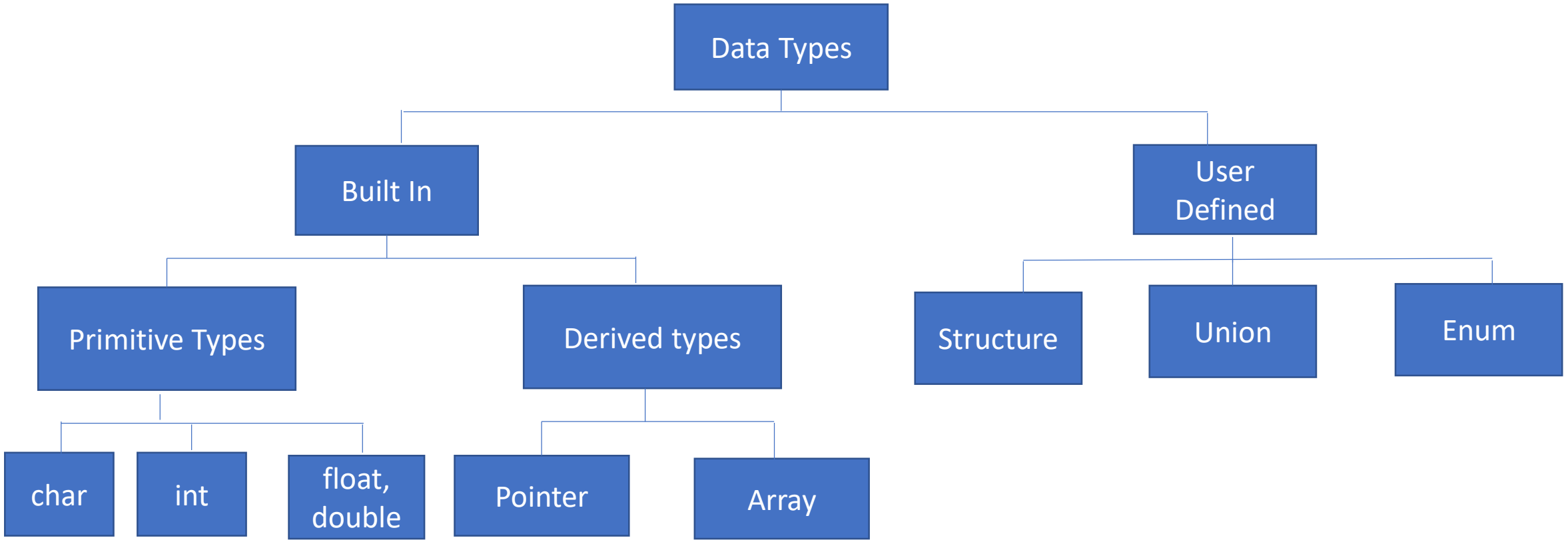
Keywords :

- Keywords are predefined words treated specially by the compiler with their standard meaning.
- These are reserved words so they cannot be used as identifiers.
- There are 27 keywords available in C language. Another 5 were added by ANSI during standardization.
- Note that few compilers also have some extra words reserved.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Data Types :

- As the name says, Data type specifies the type of data to be stored in its corresponding variable.
- Data types are classified into built in and User defined types.



Variables and constants :

- **Variables** : A variable is a storage place that has some memory allocated to it. It is used to store some form of data. Using the data types mentioned in previous slide, we can declare the variables.
- A typical variable declaration is as follows :

```
data_type variable_name;   or   data_type variable_name = value;
int num;
num = 10;
double d_num = 1.2;
char ch = ‘A’;
```

The above examples are some ways to declare and initialise the variables.

- Different types of variables require different amounts of memory. The data type of a variable represents the amount of memory assigned to it. The size of each variable or constant can be checked with sizeof() operator.
- **Constants** : Fixed data values are said to be constants.

```
12, -45, 0, 2.3, 76.9, 1.23456e+2, 'A', "Sunbeam", etc.
```

Format Specifiers :

- Format specifiers : It is a way of telling the compiler what type of data is stored in its corresponding variable, while printing the output using printf() as well as taking the input using scanf().
- char - %c
- int - %d, %i
- Unsigned int - %u
- long int - %ld
- short int - %hd
- float - %f
- Float in scientific notation - %e %E
- double - %lf
- unsigned long - %hu
- unsigned short - %hu
- string type - %s
- Pointer type - %p

sizeof operator :

- sizeof is a compile time operator.
- It is used to calculate the size of its operand.
- It returns the size of the variable in bytes.
- When sizeof is applied with datatypes, it simply return the amount of memory allocated to that datatype.
  - Int num = 10;
  - Printf(“%d”,sizeof(num));
    - Output is 4. as the size of integer datatype is 4 bytes

- When sizeof is applied with expressions it returns the size of the expression.
  - Char ch = ‘A’;
  - Printf(“%d”,sizeof(ch)); // returns 1
  - Printf(“%d”,sizeof(‘A’)); // returns 4
  - printf(“%d”,sizeof(num+ch)); // returns 4

printf() :

- printf() is a library function to send formatted output to the screen. The function prints the string inside quotations.
- To use printf() in our program, we need to include stdio.h header file using the #include<stdio.h> statement.
- Format : printf(“<format string>”, list of variables);

  It should have one format specifier for each value listed and the types should match.

  printf(“int var = %d float var = %.2f”,num, f_num);
- Arbitrary strings can also be printed using printf.

  Example : printf(“Hello Everyone, Welcome to Sunbeam!”);

Escape Sequence :

- \n - New line - Moves cursor to the beginning of next line.
- \t – Horizontal tab - Moves the cursor horizontally 8 characters ahead.
- \r - Carriage return - Moves the cursor to the beginning of current line.
- \b - Backspace - Moves cursor one position to the left of its current position.
- \f - Form Feed - Moves cursor to the beginning of next page (Used in context of printer).
- \a -Alert - Alerts by Generating beep.

- \’ - Single quote mark - prints ’
- \” - Double quote mark - prints”
- \\- Backslash Character - Prints\

Thank You