

# Values and Data Types

## Identifiers

- Used to name variables, classes, functions, structures and many other stuff
- Rules to name an identifier:
  - Can contain alphabets, numbers and underscore (\_)
  - Should start with a letter
  - Should not be a keyword
  - Case-sensitive

## Escape Sequences

Special characters used in printing, that perform a different function than the specified character.

- \n - print newline
- \t - print tab space
- \' - print single quote
- \" - print double quote
- \\ - print backslash

## Basic Primitive Data Types

- int: For integers (4 bytes)
- float: Low precision floating point numbers (4 bytes)
- double: High precision floating point numbers (8 bytes)
- char: Single character (1 byte)

## Variables

### Declaration

```
datatype variableName;  
E.g.: int i;
```

### Initialization

```
datatype variableName = value;  
E.g.: int i = 10;
```

### Assignment

```
variableName = value;  
E.g.: i = 23;
```

## Values (Literals)

### Integer Literals

Basic integers without any decimal places.  
E.g.: 69, -420, 128, 65535

### Floating point Literals

Real numbers which can be expressed in scientific form too.

a  
E.g.: 535.68, 1.602E-19

### Boolean Literals

There are only two Boolean values possible: True or False. However, unlike other languages, there are no exclusive values reserved for True and False. Instead:

- 0, empty array or the keyword null represents False
- 1 or any other number except zero represent True

### Character Literals

Character literals consist of any single character within single quotes. Escape sequences are taken as one single character.

E.g.: 'A', '3', '#', '\n'

### String Literals

String literals are a collection of one or more character literals enclosed in double quotes.

E.g.: "Hello", "12340##\$", "Next\nLine"

### The null Literal

The null keyword is a special type of literal which represents the absence of an address in a composite data type. It signifies that the variable does not point towards any address.

## Keywords

Keywords are reserved words by the compiler that have a special meaning, and therefore cannot be used as identifiers. There are 32 keywords in C.

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

## The scanf() statement

Used to accept a value from the user.

Syntax:

```
scanf(formatString, variablePointers);
```

**formatString:** a string with the format specifiers for the values that have to be accepted

**variablePointers:** pointers to the variables to store the values in

Example: if an integer value has to be accepted to the variable num, then the statement would be:

```
scanf("%d", &num);
```

### NOTE:

1) More than one value of the same datatype can be taken from a single statement

```
scanf("%d%d", &m, &n);
```

2) Be aware while taking characters as input, because the space and newline used to separate input is taken as a character. To avoid that, include the newline or space in the format string. For example:

```
scanf("%c %c", &a, &b);           or  
scanf("%c\n%c", &a, &b);
```