



M1:  
Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

# Introduction to Programming-III

Introduction to C Programming-I

Course: BTech in CSE  
Course Name: Programming for Problem Solving  
Course Code:  
Semester: II  
Session: 2019-20



Mr. Joynath Mishra  
Assistant Professor (Guest)

Department of Computer Science and Information Technology

Mahatma Gandhi Central University  
Bihar, INDIA

April 23, 2020



# Outline

## M1: Introduction to C Programming-I

Mr. J. Mishra  
MGCI, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

- 1 *Objectives*
- 2 *Introduction*
- 3 *Generalized Flow of Language*
- 4 *Program Structures*
- 5 *C Language Tokens*
  - Identifier
  - Keywords
  - Constants
  - Strings
  - Special Characters
  - Operators
- 6 *Exercise*
- 7 *References*



# Objectives

M1:

Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

## Objectives

- Study on C language tokens
- Study on C programming structure



# Introduction

M1:

Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

- It was developed by Dennis Ritchie at AT&T Bell Labs, the USA between 1969 and 1973[1].
- It was a UNIX OS developed project named as Basic Combined Programming Language(BCPL, called as B language) project at MIT, USA.
- It fill up the gap between low level to high level programming(OOPs).
- It follows divide and conquer mechanism to solve a problem, hence it is modular programming (top-to-bottom). Whereas, Object Oriented Programming(OOPs) follows procedural structure(button-up).
- It can program in register and variables both.
- Hardware level programming is created by using C language.
- LINUX kernel is developed in C.
- Linux OS, PHP, and MySQL are written in C, whereas C has written in assembly language.
- Programs written in C are efficient and fast.
- C does not provide Object Oriented Programming (OOP) concepts.
- C works as low level language as well as high level language, hence it is called as middle level language.



# Generalized Flow of Language

M1:

Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

- Language Tokens
- Conditional Operation
- Iteration Condition
- Function Creation
- Specialized Tools Creation
- File Handling

## C Language Sections

- Documentations (Documentation Section at top of any program; by using `\\` or `\\* ...*\\`)
- Preprocessor Statements (Link Section; defined by `#`)
- Global Declarations (Definition Section)
- The `main()` function( Local Declarations and Program Statements & Expressions)
- User Defined Functions

## I/O Management

- End-user interaction is performed by buffer stream files.
- Standard input, output and error file are responsible to make such interaction.
- ***stdin*** associated with user input stream, ***stdout*** associated with ***stdin*** and corresponding output, ***stderr*** associated with output stream and error message.



# Program Structures

M1:

Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

## Source Code (C Language)

```
1 #include "stdio.h" /*Preprocessor search file and Load header
   file into memory*/
2 int main() /*Execution starts from main() and it is
   called as user defined library function*/
3 {
4 printf("Welcome to C Language World!!!");
5 return 0; /*Any function should return a value to indicate
   the successful completion of it*/
6 }
```

## Execution (LINUX Terminal)

```
Compile: "cc example.c -o example"
View Machine Code(not necessary): "xxd example"
Execute: "./example"
Status (return from last execution): "echo $?"
```

## Output

```
Welcome to C Language World!!!
```



# C Language Tokens[2],[3]

M1:  
Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

## Tokens

- Identifier
- Keywords
- Strings
- Constants
- Special Characters
- Operators



# Identifiers

M1:

Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

## Identifiers

- It indicates names given to a program element such as variable, array, function(user defined).
- The first character in an identifier must be an alphabet or an underscore and can be followed only by any number alphabets, or digits or underscores.
- They must not begin with a digit.
- Uppercase and lowercase letters are distinct. That is, identifiers are case sensitive.
- Commas or blank spaces are not allowed within an identifier.
- Keywords cannot be used as an identifier.
- Identifiers should not be of length more than 31 characters.
- Identifiers must be meaningful, short, quickly and easily typed and easily read.
- It is two types as internal(*local*) and external(*global*) identifier.

## Identifiers

- `int price;`
- `double interest;`





# Identifiers (Contd...)

## Data Types

M1:  
Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

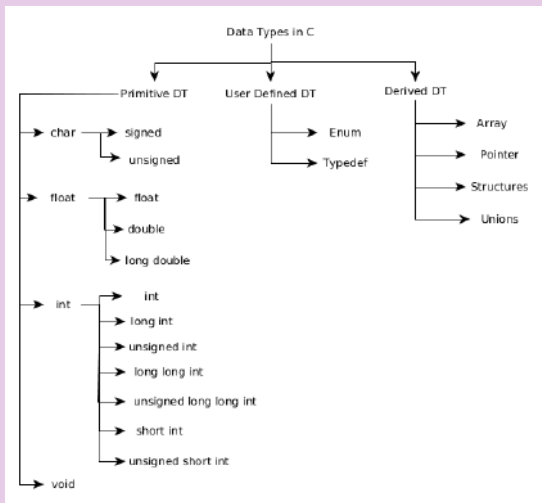


Figure 1: Different Data Types used in C Language.



# Identifiers (Contd...)

## Identifier Range, Format Specifier and Variable Declaration

Type	Range		Bytes	Represents
	From	To		
char/short	-128	127	1	characters
unsigned char	0	255	1	characters
int	-32,768	32,767	2	whole numbers
unsigned int	0	65,535	2	whole numbers
long	-2,147,438,648	2,147,438,647	4	whole numbers
unsigned long	0	4,294,967,295	4	whole numbers
float	$3.4 \times 10^{-38}$	$3.4 \times 10^{38}$	4	fractional numbers
double	$1.7 \times 10^{-308}$	$1.7 \times 10^{308}$	8	fractional numbers
long double	$3.4 \times 10^{-4932}$	$3.4 \times 10^{4932}$	10	fractional numbers

Figure 2: Different Identifier Data Type, Range and Memory Occupation.

Table 1: C Language Format Specifiers.

%c	a single character
%d or %i	a decimal integer
%f	a floating point number
%h	a short integer
%o	a octal number
%x	a hexadecimal number
%u	a unsigned integer
%ld	a long data type
%lf	a double
%s	a string

Table 2: Automatic variable declaration and initialization.

<code>int x;</code>	<code>x = 5;</code>	<code># define PI 3.14</code>
<code>float x;</code>	<code>x=3.7E12;</code>	<code>int x=5;</code>
<code>int x;</code>	<code>x = 012;</code>	<code>float x=3.7E12;</code>
<code>int x;</code>	<code>x = 0x12;</code>	<code>int x = 012;</code>
<code>char ch;</code>	<code>ch='T'</code>	<code>int x = 0x12;</code>
<code>char name[20];</code>	<code>name="JNM"</code>	<code>char ch='T';</code>
<code>bool x;</code>	<code>x=false;</code>	<code>char name[20]="JNM";</code>
		<code>bool x=false;</code>



# Keywords

M1:

Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCI, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

## Keywords

- Some words are reserved in C compiler to make sense of particular syntax, called as *keywords* or *reserved words*.
- These words cannot be used as identifier.

Table 3: C Language Keywords

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



# Keywords (Contd...)

M1:  
Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

## Example

```
1 #include<stdio.h>
2 int main() //int is a keyword
3 {
4 float a, b, s; //float is a keyword
5 printf("Please enter two values:");
6 scanf("%f%f", &a,&b);
7 s=a+b;
8 printf("Sum of two numbers = %f", s);
9 return 0; //return is a keyword
10 }
```

## Output

```
Please enter two values:5 6
Sum of two numbers = 11.000000
```



# Constants

M1:

Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCU, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

**Constants**

Strings

Special Characters

Operators

Exercise

References

## Primary Constants

Table 4: C Language Primary Constants

Constants	Description
<code>\a</code>	beep sound
<code>\b</code>	backspace
<code>\f</code>	form feed
<code>\n</code>	new line
<code>\r</code>	carriage return
<code>\t</code>	horizontal tab
<code>\v</code>	vertical tab
<code>'</code>	single quote
<code>"</code>	double quote
<code>\\</code>	backslash
<code>\0</code>	null

## Secondary Constants

- Array
- Pointer
- Structure
- Union
- Enum



## Constants (Contd...)

M1:  
Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

**Constants**

Strings

Special Characters

Operators

Exercise

References

### Example

```
1 #include<stdio.h>
2 const int EMPID = 150;
3 int main()
4 {
5     const double salary = 20000;
6     printf("Employee Id = %d", EMPID);
7     printf("Employee salary = %5.2f", salary);
8     return 0;
9 }
```

### Output

```
Employee Id = 150
Employee salary = 20000.00
```



# Strings

M1:  
Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCI, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

## Example

```
1 #include <stdio.h>
2 int main()
3 {
4     char name[20];
5     printf("Enter your name: ");
6     scanf("%s", name);
7     printf("Your name is %s.", name);
8     return 0;
9 }
```

## Output

```
Enter your name: mishra
Your name is mishra
```



# Special Characters

M1:  
Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCU, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

Special characters are not use for identifier naming.

## Special Characters

Table 5: C Language Special Characters

comma(,)	<	>	dot(.)	underscore(_)
parentheses()	parentheses()	semi-colon(;)	\$	colon(:)
percent( % )	square bracket[]	square bracket[]	hash(#)	?
single quote(')	ampersand( & )	bracket open{}	bracket close{}	double quote("")
power( )	!	multiply(*)	front slash(/)	pipe(  )
minus(-)	\	tilde(~)	plus(+)	

## White Space Characters

Blank space, newline, horizontal tab, carriage, return and form feed.

## Control Characters

The characters which don't occupy any printing position, called as control characters such as NUL, DEL.





# Operators

M1:

Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

## Type of Operators

Operators are important to form any expression and to perform any arithmetic and logical operations.

- Arithmetic Operators ( $++$ ,  $-$ ,  $*$ ,  $/$ ,  $\%$ ,  $+$ ,  $-$ )
- Relational Operators ( $>$ ,  $>=$ ,  $<$ ,  $<=$ ,  $==$ ,  $!=$ )
- Logical Operators ( $\&\&$ ,  $\|\|$ ,  $!$ )
- Assignment Operators ( $=$ ,  $+=$ ,  $-=$ )
- Pointer Operators ( $*$ )
- Bitwise Operators ( $\ll$ ,  $\gg$ )
- Special Operators (comma( $,$ ),  $\text{sizeof}()$ )
- Conditional Operators ( $?:$ )



# Exercise

## M1: Introduction to C Programming-I

Mr. J. Mishra  
MGCUB, INDIA

### Objectives

### Introduction

### Generalized Flow of Language

### Program Structures

### C Language Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

### Exercise

### References

- What are the rules to define identifier in C language?
- What do you mean by reserved words in C and what are these?
- Write a program in C language to create a critical beep sound in system.
- What are the types of constants in C language?
- Describe string manipulation in C language.
- What do you mean by special character, white space characters and control characters?
- What are the input separation mechanism at time of input insertion in executed C code?



# References I

M1:

Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References



D. M. Ritchie, "The development of the c language," *ACM Sigplan Notices*, vol. 28, no. 3, pp. 201–208, 1993.



S. Jain, *Programming and Problem Solving through C Language Design*. BPB Publications, 2003.



K. Yashwant, "Let us c," *Array and pointers, 7th edition, BPB publication*, 1999.



M1:  
Introduction to  
C  
Programming-I

Mr. J. Mishra  
MGCUB, INDIA

Objectives

Introduction

Generalized Flow  
of Language

Program  
Structures

C Language  
Tokens

Identifier

Keywords

Constants

Strings

Special Characters

Operators

Exercise

References

Get in touch via...



+91 9046174189



jaynath4025@gmail.com

*Thank You...*