C Programming Language

Saikat Hudait

Student Of Computer Science Engineering , B.tech

Content

- Introduction of C
- Structure of a C-language program
- Write your first C program
- Input and output concepts
- C compiler
- C basic data types
- Characteristics of C
- Applications of C

Introduction of C

- C is a procedural programming language.
- It was developed by Dennis Ritchie in the year 1972.
- It was mainly developed as a system programming language to write an operating system.

C is a case-sensitive language.

Conti…

- The Unix operating system and virtually all Unix applications are written in the C language.
- C is a successor of B language which was introduced around 1970
- The language was formalized in 1988 by the American National Standard Institute (ANSI).
- By 1973 UNIX OS almost totally written in C.
- Today C is the most widely used System Programming Language.

Conti…

- C evolved from two previous languages, BCPL (Basic Combined Programming Language) and B.
- BCPL developed in 1967 by Martin Richards as a language for writing O.S and compilers.
- Ken Thompson modeled many features in his language, B, after their counterparts in BCPL, and used B to create an early versions of UNIX operating system at bell Laboratories in 1970 on a DEC PDP-7 computer.

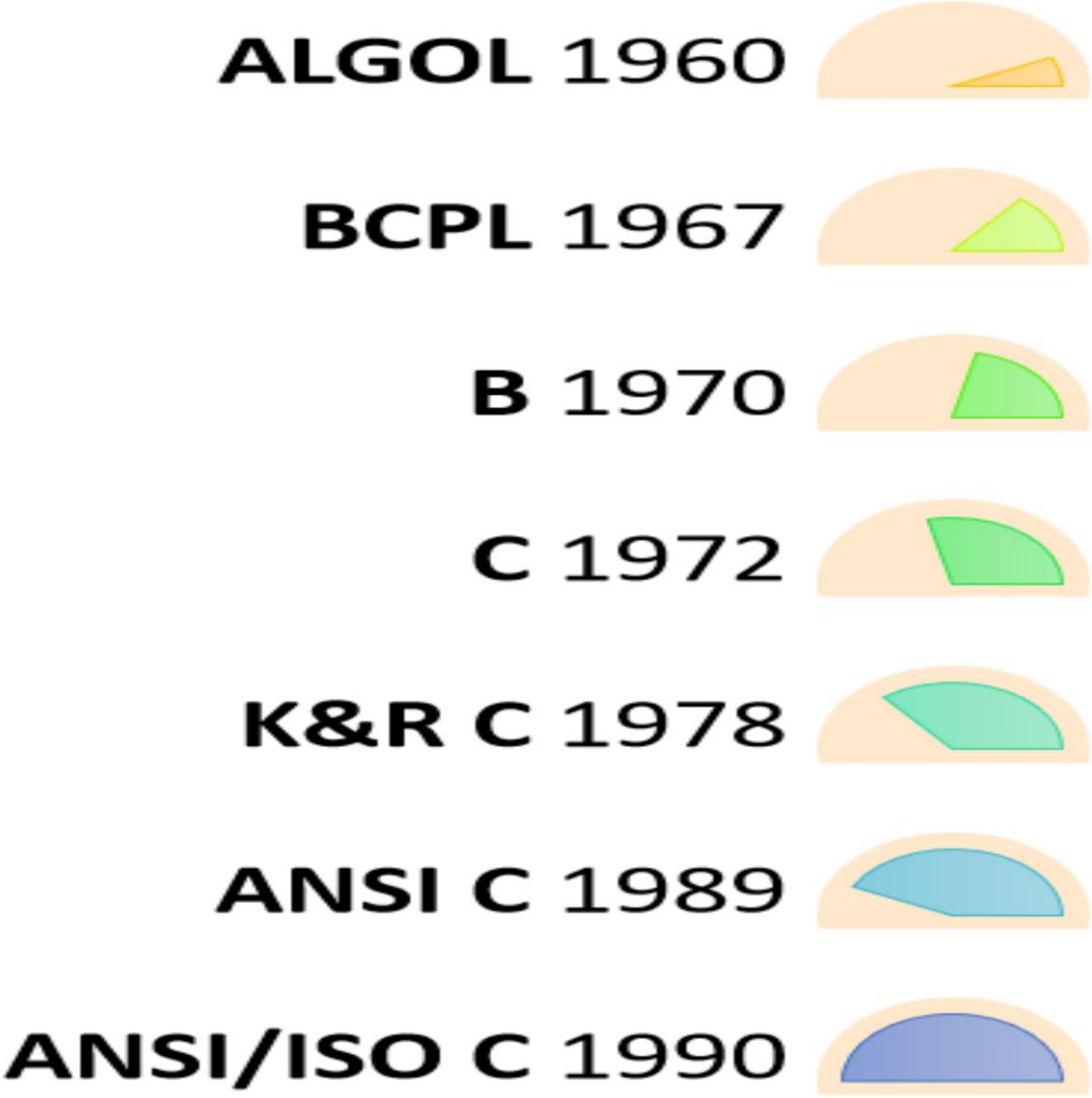
Conti…

- Both BCPL and B were type-less languages: the only data type is machine word and access to other kinds of objects is by special operators or function calls.
- The C language developed by Dennis Ritchie at Bell Laboratories and was originally implemented on a DEC PDP-11 computer in 1972.
- It was named C for new language (after B).
- Initially, C used widely as the development language of the UNIX OS.

Conti…

- In 1983, the X3J11 technical committee was created under the American National Standards Institute (ANSI) Committee on Computer and Information Processing (X3) to provide an unambiguous and machine-independent definition of the language and approved in 1989, called ANSI C.
- Then, the document is referred to as ANSI/ISO 9899:1990.
- The second edition of Kernighan and Ritchie, published in 1988, this version called ANSI C, then used worldwide.
- Today, almost all new major OS are written in C including Windows.

Conti…



ALGOL 1960

BCPL 1967

B 1970

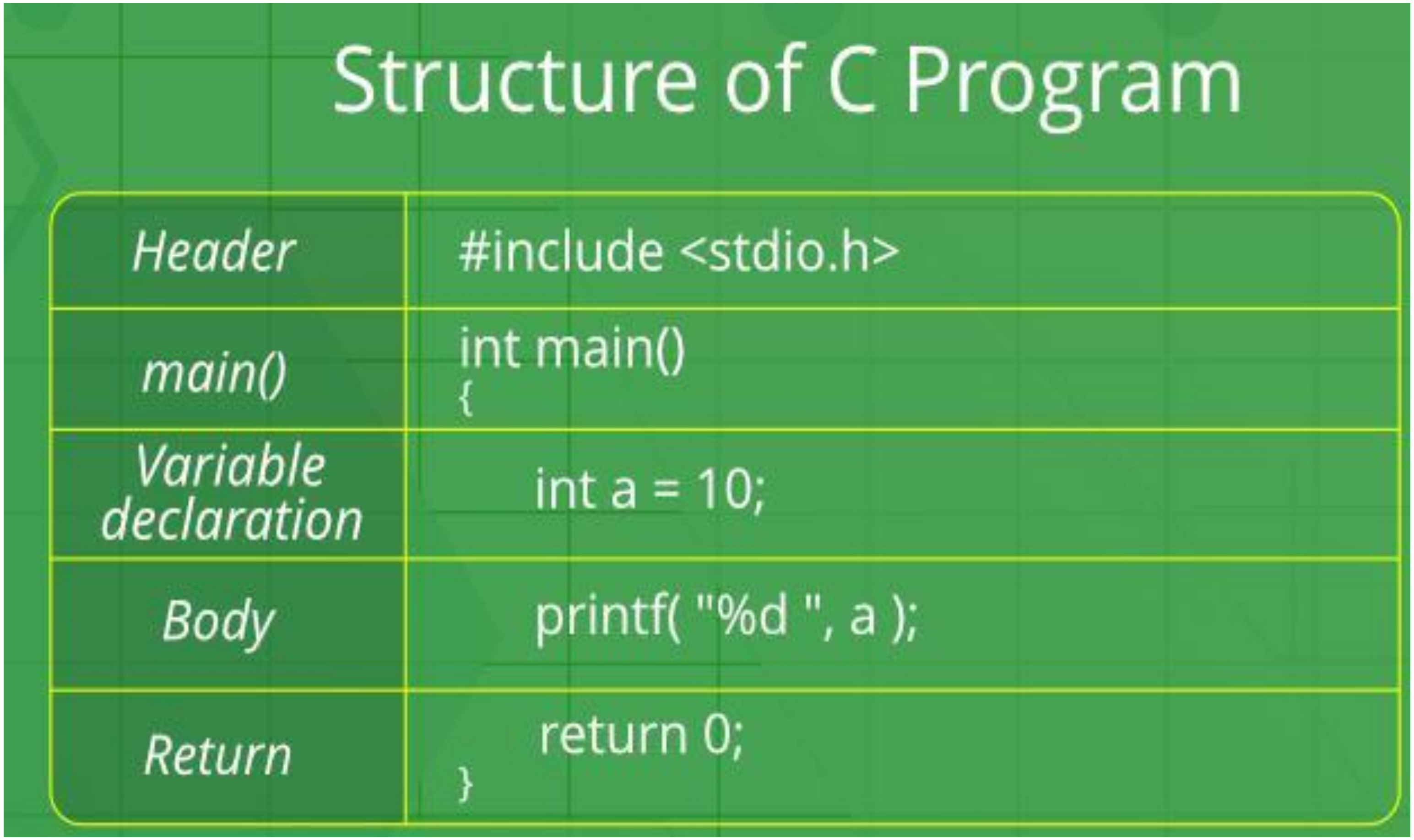
C 1972

K&R C 1978

ANSI C 1989

ANSI/ISO C 1990

Structure of a C-language program



Structure of C Program

Header	#include <stdio.h>
main()	int main() {
Variable declaration	int a = 10;
Body	printf("%d ", a);

Return	return 0;
	}

Fig. 1 Structure of a C Program

Component of C Program

- **Header Files:** The first and foremost component is the inclusion of the Header files in a C program.
- A header file is a file with extension .h
- Which contains C function declarations and macro definitions to be shared between several source files.

Example of Header files:

stddef.h – Defines several useful types and macros.

stdint.h – Defines exact width integer types.

stdio.h – Defines core input and output functions

stdlib.h – Defines numeric conversion functions, pseudo-random network generator, memory allocation

string.h – Defines string handling functions math.h – Defines common mathematical functions

Conti…

Main Method: The next part of a C program is to declare the main() function.

e.g. main()

 {}

Variable Declaration: The next part of any C program is the variable declaration.

-In the C program, no variable can be used without being declared.

-Also in a C program, the variables are to be declared before any operation in the function.

Conti…

Ex.

```
main()
{
int a;
```

Body of program: Body of a function in C program, refers to the operations that are performed in the functions.

Example:

```
int main()
{
int x;
printf("%d", x); . .
```

Conti…

Return Statement: The last part in any C program is the return statement.

-The return statement refers to the returning of the values from a function.

-This return statement and return value depend upon the return type of the function.

For example: if the return type is void, then there will be no return statement.

Conti…

```
int main()
{
int a;
printf("%d", a);
 return 0;
}
```

C Programming Basics to Write A C Program

C Basic commands	Explanation
#include <stdio.h>	This is a preprocessor command that includes standard input output header file(stdio.h) from the C library before compiling a C program
int main()	This is the main function from where execution of any C program begins.
{	This indicates the beginning of the main function.
/*_some_comments_*/	whatever is given inside the command “/* */” in any C program, won’t be considered for compilation and execution.
printf(“Hello_World! “);	printf command prints the output onto the screen.
getch();	This command waits for any character input from keyboard.

Conti…

return 0;	This command terminates C program (main function) and returns 0.
}	This indicates the end of the main function.

```
#include <stdio.h>
int main()
{
   /* Our first C basic program
   */ printf(“My first program ");
   getch();

}
```

Steps to Write C Programs and Get the Output

This is common to all C program

- Create
- Compile
- Execute or Run
- Get the Output

Key Points

- Each C programming statement is ended with semicolon (;) which are referred as statement terminator.
- A header file is a file with extension .h which contains C function declarations and macro definitions to be shared between several source files.
- Printf() command is used to print the output onto the screen.

- **C programs are compiled using C compilers and displays output when executed.**

Conti…

- the C programs are writen into text files with extension ".c" , Ex. First.c
- To run “c” program you need to compile that program using a C Compiler which converts your program into a language understandable by a computer.

C compiler

- Many compilers are available for executing programs written in 'C'.
- A compiler compiles the source file and generates an object file.
- A linker links all the object files together and creates one executable file.

Example : AMPC, CCS C Compiler, ch, clang, Cygwin, Digital mars, GCC compiler, MikroC Compiler, Portable C Compiler, Power C, QuickC, Ritchie C Compiler, Small-C

C Data Types

- Each variable in C has an associated data type.
- The type of a variable determines how much space it occupies in storage and how the bit pattern stored is interpreted.
- Each data type requires different amounts of memory and has some specific operations which can be performed over it.

Conti...

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	8 bytes	-9223372036854775808 to 9223372036854775807
unsigned long	8 bytes	0 to 18446744073709551615

Identifiers

- One feature present in all computer languages is the identifier.
- Identifier refers to name given to entities such as variables, functions, structures etc.
- Identifiers must be unique.
- Remember, identifier names must be different from keywords.

An identifier must start with a letter or underscore: it may not have a space or a hyphen.

Rules for Identifiers

- A valid identifier can have letters (both uppercase and lowercase letters), digits and underscores.
- The first letter of an identifier should be either a letter or an underscore.
- You cannot use keywords as identifiers.
- There is no rule on how long an identifier can be. (However, you may run into problems in some compilers if the identifier is longer than 31 characters.)

Example

Correct	Incorrect
testprogram	1testprogram /* starts with a digit */
_arrayex	%arrayex /* contains invalid character */
charAndNum	Char /* reserved word */
structure_prg	Structure prg /* contains a space */

Variables

- In programming, a variable is a container (storage area) to hold data.
- To indicate the storage area, each variable should be given a unique name (identifier).
- Variable names are just the symbolic representation of a memory location.

When a variable is defined, it is not initialized. We must initialize any variable requiring prescribed data when the function starts.

Rules for naming a variable

- A variable name can only have letters (both uppercase and lowercase letters), digits and underscore.
- The first letter of a variable should be either a letter or an underscore.

Note: You should always try to give meaningful names to variables.

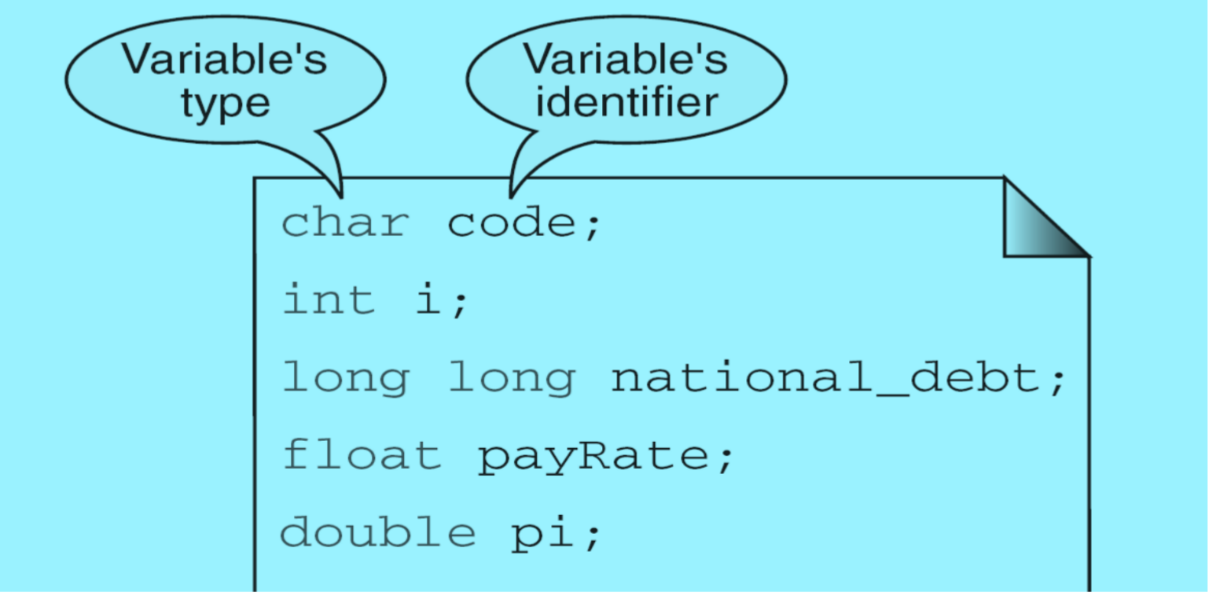
For example:

```
int number = 5; // integer variable
double number; // error
```

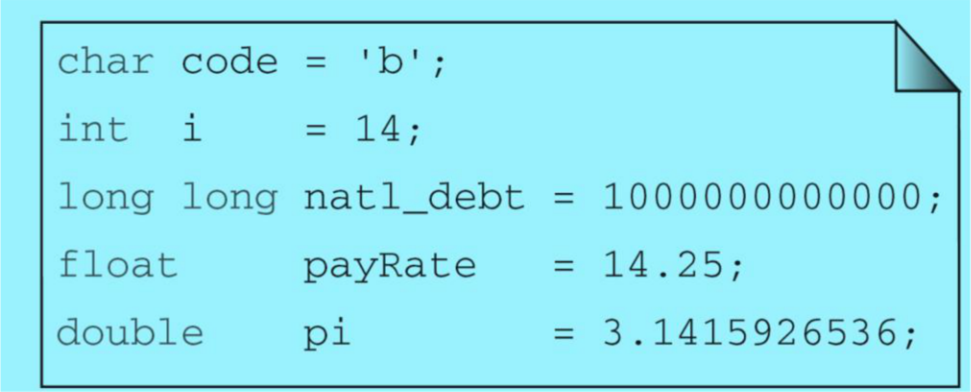
Variables

- In programming, a variable is a container (storage area) to hold data.
- To indicate the storage area, each variable should be given a unique name (identifier).
- Variable names are just the symbolic representation of a memory location.

Variable Declaration/Initialisation



```
char code;
int i;
long long national_debt;
float payRate;
double pi;
```



```
char code = 'b';
int  i    = 14;
long long natl_debt = 1000000000000;
float     payRate   = 14.25;
double    pi        = 3.1415926536;
```

Constants

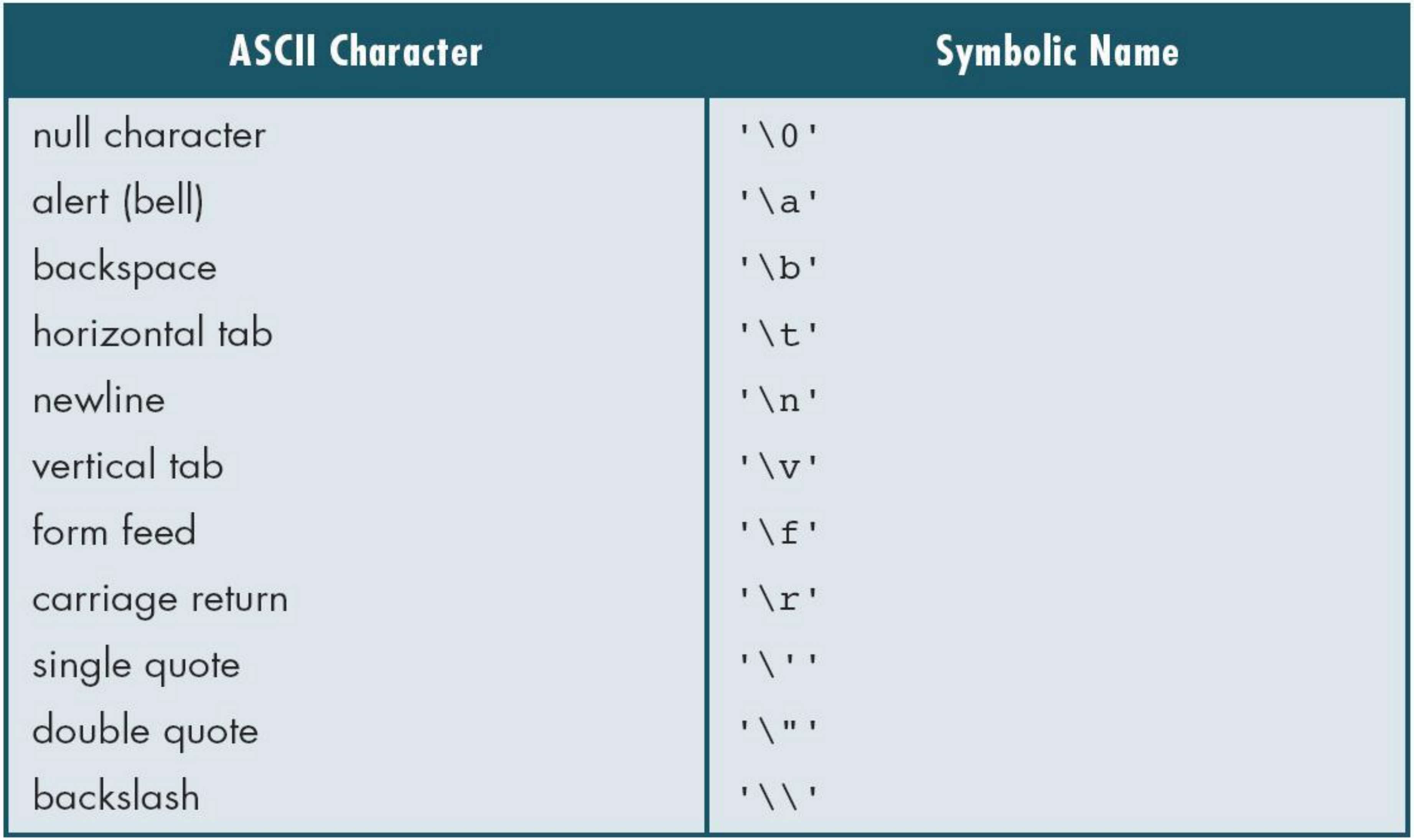
- Constants are data values that cannot be changed during the execution of a program. Ex. variables, constants have a type.

Ex: int a=10

A character constant is enclosed in single quotes.

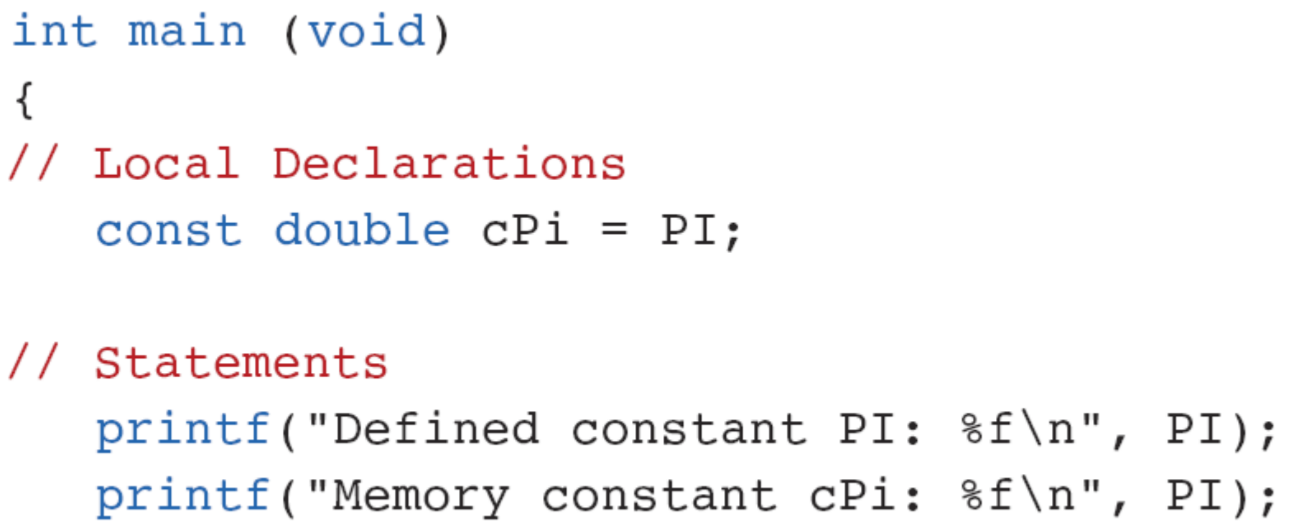
A character constant is enclosed in single quotes

Symbolic Names for Control Characters



ASCII Character	Symbolic Name
null character	`'\0'`
alert (bell)	`'\a'`
backspace	`'\b'`
horizontal tab	`'\t'`
newline	`'\n'`
vertical tab	`'\v'`
form feed	`'\f'`
carriage return	`'\r'`
single quote	`'\''`
double quote	`'\"'`
backslash	`'\\'`

Conti…



```
int main (void)
{
// Local Declarations
   const double cPi = PI;

// Statements
   printf("Defined constant PI: %f\n", PI);
   printf("Memory constant cPi: %f\n", PI);
```

Use single quotes for character constants.
Use double quotes for string constants.

C input/output

- Use scanf() function to take input from the user.
- printf() function to display output to the user.

Ex.

```
#include <stdio.h>
 int main()
 {
int test1;
printf("Enter an integer: ");
scanf("%d", &test1);
printf("Number = %d",test1);
getch();
 }
```

C OPERATORS

- An expression is any valid set of literals, variables, operators, operands and expressions that evaluates to a single value.
- This value can be a number, a string or a logical value.
- For instance a = b + c; denotes an expression in which there are 3 operands a, b, c and two operator + and =.
- A statement, the smallest independent computational unit, specifies an action to be performed.

Conti…

- In most cases, statements are executed in sequence.
- The number of operands of an operator is called its arity.
- Based on arity, operators are classified as nullary (no operands), unary (1 operand), binary (2 operands), ternary (3 operands).

C Keywords/Reserved Words

- As C is a case sensitive language, all keywords must be written in lowercase.
- Keywords are predefined, reserved words used in programming that have special meanings to the compiler.
- Keywords are part of the syntax and they cannot be used as an identifier.

Conti…

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

Comments

- Single line of comment: // comment here
- More than single line of comment or expanded: /* comment(s) here */

```
// for printf()
#include <stdio.h>
#include <string.h> // for strcpy_s() and their family

/* main() function, where program
   execution starts */
int main()
{
    /* declares variable and initializes it*/
    int i = 8;
…
```

Characteristics of C

- C is a general-purpose programming language that is extremely popular, simple and flexible.
- It is machine-independent, structured programming language which is used extensively in various applications.
- C was the basics language to write everything from operating systems (Windows and many others) to complex programs like the Oracle database, Git, Python interpreter and more.

Conti…

- The main features of C language include low-level access to memory, a simple set of keywords, and clean style.
- These features make C language suitable for system programming‘s like an operating system or compiler development.
- Many later languages have borrowed syntax/features directly or indirectly from C language.
- Ex. syntax of Java, PHP, JavaScript, and many other languages are mainly based on C language.
- C++ is nearly a superset of C language.

Applications of C

- C was initially used for system development work, in particular the programs that make-up the operating system.
- C was adopted as a system development language because it produces code that runs nearly as fast as code written in assembly language.
- Some examples: Operating Systems, Language Compilers, Assemblers, Text Editors, Print Spoolers, Network Drivers, Modern Programs, Data Bases, Language Interpreters, Utilities.

Applications of C

- 'C' language is widely used in embedded systems.
- It is used for developing system applications.
- It is widely used for developing desktop applications.
- Most of the applications by Adobe are developed using 'C' programming language.
- It is used for developing browsers and their extensions. Google's Chrome is built using 'C' programming language.

Conti…

- It is used to develop databases. MySQL is the most popular database software which is built using 'C'.
- It is used in developing an operating system. such as Apple's OS X, Microsoft's Windows, and Symbian are developed using 'C' language. It is used for developing desktop as well as mobile phone's operating system.
- It is used for compiler production.
- It is widely used in IOT applications.

Thank You

Contact:
Email:
saikathudait2001@gmail.com