

Overview of 'C' Programming Language

Sisira Kumar Kapat

Department of Computer Science & Engineering,
UCP Engineering School,
Berhampur, Orissa

Contents

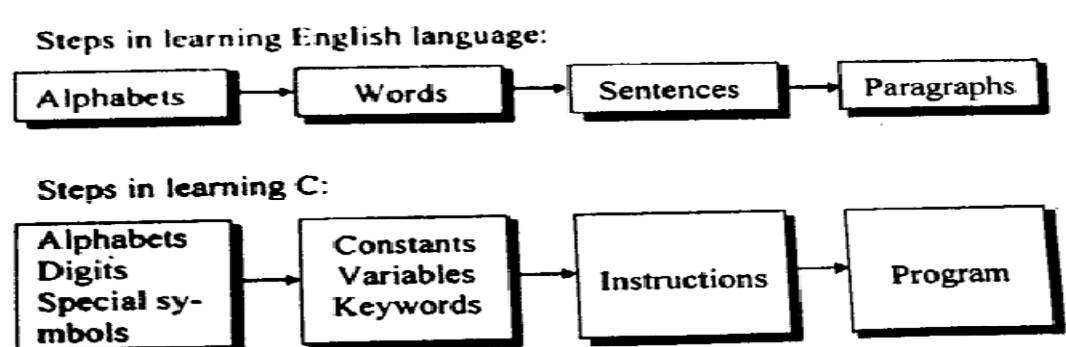
- Constants, Variables and Data types in C
- Managing Input and Output operations.
- Operators, Expressions, Type conversion & Typecasting
- Decision Control and Looping Statements (If, If-else, If-else-if, Switch, While, Dowhile, For, Break, Continue & Goto)
- Programming Assignments using the above features.


Introduction

- C – Programming
 - 1972, AT&T's Bell Lab, USA
- Structure Oriented language

Elements of C

- C character set
 - Alphabets
 - Digits
 - Special Characters
- Escape sequence
- Trigraph characters
- Delimiters
- Key words
- Variables and Constants
- Data types



- 
- Constants: An entity that doesn't change during the execution of program.
 - Variables : Named memory location, where some data can be stored and the value may vary during program execution.
 - Data type : The type of data used in the program.
 - Format Specifier

Escape sequence

Escape Sequence	Meaning
<code>\b</code>	backspace
<code>\a</code>	bell(alert)
<code>\r</code>	carriage return
<code>\n</code>	newline
<code>\f</code>	form feed
<code>\0</code>	null
<code>\v</code>	vertical tab
<code>\t</code>	Horizontal tab
<code>\\</code>	backslash

Constants

- An entity that doesn't change during the execution of program.
- Primary constants
 - Integer constant
 - Real constant
 - Character constant
- Secondary constants
 - Array
 - Pointer
 - Structure
 - Union etc.

Primary Constant

- Integer constant
 - Positive or negative numerical values.
 - Must not have a decimal point.
 - No comma or blanks allowed within an integer constant.
 - Allowable range is -32768 to 32767.
 - Example: 6453, 3248 etc.
- Real constant
 - A positive or negative decimal fraction.
 - At least one digit.
 - No comma or blanks allowed within an integer constant.
 - Allowable range is -3.4e38 to 3.4e38.
 - Example: 0.5, 4000.0, 5597. etc.

Primary Constant (cont...)

- Character constant
 - A single alphabet or a single digit or a single special symbol enclosed within single inverted comma.
 - Example: 'A', 'M', '7', '#' etc.
- String constant
 - Zero, one or more than one character enclosed within double quotes.
 - At the end of the string, \0 is automatically placed by the compiler.
 - Example: "India", "3624", "", "I" etc.

Variables

- Named memory location, where some data can be stored and the value may vary during program execution.
- Rules for constructing variables
 - Combination of alphabets, digits or underscore.
 - The first character must be an alphabet or underscore.
 - No comma or blanks are allowed within variable name.
 - No special symbol other than underscore is allowed.
 - Example: simp_int, age, name, tot_mark etc.
- The type of variable must be declared at the beginning of the program.
- Example,
int a, num, age;
float sim_interest;

Reserved/ Key words

- Certain words that are reserved for doing special task.
- They have standard, predefined meaning in C
- Always written in lower case letters
- 32 key words are available 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

Summary sheet

Name of cont/var	Data type used	Size	Range	Format specifier
Character	Char	1 byte	-128 to 127	%c
String	Char[num]	-	-	%s
Integer	int	2 bytes	-32768 to 32767	%d
Real	Float	4 bytes	-3.4e38 to 3.4e38	%f
Double	double	8 bytes	-1.7e308 to 1.7e308	

Input/ Output

- Input

- Statement is used: `scanf();`

- Example: `scanf(“%d”, age);`

- Here, ‘age’ is the name of variable, where the number (%d) entered through keyboard will be stored.

- Output

- Statement used: `printf();`

- Example1: `printf(“this is a test”);`

- Here the statement ‘this is a test’ will be displayed.

- Example2: `printf(“Your age is %d”,age);`

- Here the value stored in variable ‘age’, will be displayed.

A program with an output statement

Program

```
#include<stdio.h>
void main()
{
    printf("This is the first C program");
}
```

Output

This is the first C program

A program with an output statement

Program

```
#include<stdio.h>
void main()
{
    printf("This is the first C program");
}
```

Output

This is the first C program

A program with an input statement

Program

```
#include<stdio.h>
void main()
{
    int number;

    scanf("%d",&number);

    printf("You entered %d",number);
}
```

Output

```
1503
You entered 1503
```


A program with an input statement

Program

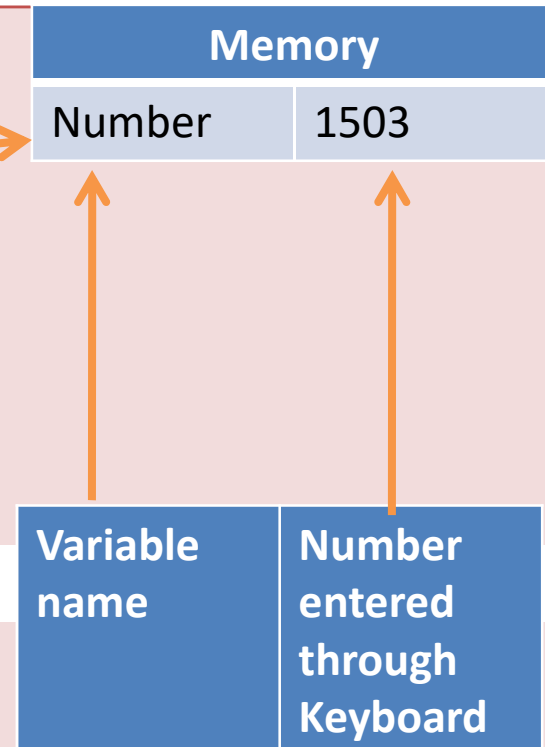
```
#include<stdio.h>
void main()
{
    int number;

    scanf("%d",&number);

    printf("You entered %d",number);
}
```

Output

```
1503
You entered 1503
```



A program with an input statement

Program

```
#include<stdio.h>
void main()
{
    int number;

    scanf("%d",&number);

    printf("You entered %d",number);
}
```

Output

1503

You entered 1503

A program with an input statement

Program

```
#include<stdio.h>
void main()
{
    int number;

    scanf("%d",&number);

    printf("You entered %d",number);
}
```

Output

1503

You entered **1503**

Comments used in C

- The line written using comment, will not be executed.
- Single line comment
 - Symbol used: `//`
 - Example:
`//this is a test`
- Multiple line comment
 - Symbol used: `/*....*/`
 - Example:
`/* this is a test*/`

Operators

- Type-1
 - Arithmetic operator (+, -, *, /)
 - Relational operator (<, <=, >, >=)
 - Logical operator (&&, ||, !)
 - Assignment operator (=)
 - Increment/Decrement operator (++ , --)
 - Sizeof operator (sizeof())
 - Bitwise operator (&, |, ~, <<, >>, ^)
 - Other operator
- Type-2
 - Unary operator: operates on a single operand. (++ , --)
 - Binary operator: operates on two operands. (+, -, *)
 - Ternary operator: operates on three operands. (? :)

Operators

- Arithmetic operator
 - Operators used for arithmetic operations (+, -, *, /, % etc)
 - Example: $a+b$, $a-b$ etc.
- Relational operator
 - Used to deduce relation between two variables, such as $<$, $>$, $<=$, $>=$ etc.
 - Example: $a<b$, $a>=b$ etc.
- Logical operator
 - Used when we use the word 'and', 'or', 'not' etc.
 - Example: $a \ \&\& \ b$, $a \ || \ b$ etc.

Example of Arithmetic operator

Program

```
/*A program to demonstrate the example of arithmetic operator  
Written on 27th April 2020*/
```

```
//A program to add the given two numbers
```

```
#include<stdio.h> //header file  
void main()      //main function starts  
{  
    int a=5, b=7; //variable declaration  
  
    printf("The sum is %d", a+b); //output statement  
}
```

Output

```
The sum is 12
```

Control statement statement

- Decision
- Loop
- Switch
- Jump

Decision Control

- if
- if-else
- Nested if-else

If-statement

Syntax:

```
if(condition)
{
    statement1;
    statement2;
    .....
    statement n;
}
```

example:

```
if(number<=0)
{
    printf("Not a natural number");
    printf("natural number starts from 1");
    printf("natural number ends at infinity");
}
```

Curly brace {} required for multiple statements

Syntax:

```
if(condition)
    statement;
```

Example:

```
if(number<=0)
    printf("not a natural number");
```

Curly brace {} is not required for single statement

Example of relational, logical operator, conditional statement

Program

```
//A program to find largest among three numbers

#include<stdio.h>
void main()
{
    int num1, num2, num3;

    printf("Enter three numbers:");
    scanf("%d %d%d",&num1,&num2,&num3);

    if(num1>num2 && num1>num3)
        printf("num1 is largest");
    if(num2>num3 && num2>num1)
        printf("num2 is largest");
    if(num3>num1 && num3>num2)
        printf("num3 is largest");
}
```

Output

```
Enter three numbers: 15 23 16
Num2 is largest
```

Example of relational, logical operator, conditional statement(cont...)

Program

```
//A program to find largest among three numbers

#include<stdio.h>
void main()
{
    int num1, num2, num3;

    printf("Enter three numbers:");
    scanf("%d %d%d",&num1,&num2,&num3);

    if(num1>num2 && num1>num3)
        printf("%d is largest", num1);
    if(num2>num3 && num2>num1)
        printf("%d is largest", num2);
    if(num3>num1 && num3>num2)
        printf("%d is largest", num3);
}
```

Output

```
Enter three numbers: 15 23 16
23 is largest
```

if-else statement

- Syntax:

```
if(condition)
{
    true part;
}
else
{
    false part;
}
```

- example:

```
if(number<=0)
{
    printf("Not a natural number");
}
else
{
    printf("natural number");
}
```

Example of if-else statement

Program

```
//A program to find largest among three numbers

#include<stdio.h>
void main()
{
    int mark, result;

    printf("Enter mark:");
    scanf("%d",&mark);

    if(mark>=35)
        printf("\npass");
    else
        printf("\nfail");
}
```

Output

```
Enter mark: 45
pass
```

Nested if-else statement

Syntax:

```
if(condition1)
{
    statements;
}
else if (condition2)
{
    statements
}
....
....
else
{
    statements
}
```

Example:

```
if(mark >=60)
{
    printf("first division");
}
else if(mark>=50 && mark<60)
{
    printf("second division");
}
else
{
    printf("fail");
}
```

Example of nested if-else statement

Program

```
//A program to find largest among three numbers

#include<stdio.h>
void main()
{
    int mark, result;

    printf("Enter mark:");
    scanf("%d",&mark);

    if(mark>=35)
        printf("\npass");
    else
        printf("\nfail");
}
```

Output

```
Enter mark: 45
pass
```


Example: Addition of first 10 natural number (cont...)

- C Program

```
#include<stdio.h>
void main()
{
    int num=1, sum=0;
    ab: sum=sum+num;
    if(num<10)
    {
        num=num+1;
        goto ab;
    }
    else
        printf("The sum of first 10 natural number is %d", sum);
}
```

 DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra... —

```
The sum of first 10 natural number is 55_
```

Loop Control

- while
- do-while
- for

Support to a loop for condition

- Variable initialization
 - Example
`int num=1;`
- Condition
 - Example
`while(num%2==0)`
- Increment or decrement
 - Example
`num++;`

While loop

Syntax:	Example:
<pre>variable initialize; while(condition) { statements; inc/dec; }</pre>	<pre>num=1; while(num<100) { printf("natural number"); num++; }</pre>

do-while loop

Syntax:	Example:
<pre>variable initialize; do { statements; inc/dec; } while(condition);</pre>	<pre>num=1; do { printf("natural number"); num++; } while(num<100);</pre>


For loop

Syntax:	Example:
<pre>for(initialize;condition;inc/dec) { statements; }</pre>	<pre>for(num=1;num<100;num++) { printf("natural number"); }</pre>

Example1: Addition of first 10 natural number (cont...)

- C Program

```
#include<stdio.h>
void main()
{
    int num=1, sum=0;
    ab: sum=sum+num;
    if(num<10)
    {
        num=num+1;
        goto ab;
    }
    else
        printf("The sum of first 10 natural number is %d", sum);
}
```

 DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra... —

The sum of first 10 natural number is 55_

Example2: print natural numbers up to 100

While loop

```
#include<stdio.h>
void main()
{
    int num=1;
    while(num<=100)
    {
        printf("%d \t",num);
        num++;
    }
}
```

Do-while loop

```
#include<stdio.h>
void main()
{
    int num=1;
    do
    {
        printf("%d \t",num);
        num++;
    } while(num<=100);
}
```

For loop

```
#include<stdio.h>
void main()
{
    int num;
    for(num=1;num<=100;num++)
    {
        printf("%d \t",num);
    }
}
```

Output

1 2 3 4 5 6 100

Example3: print even numbers up to 100

While loop

```
#include<stdio.h>
void main()
{
    int num=1;
    while(num<=100)
    {
        if(num%2==0)
            printf("%d \t",num);
        num++;
    }
}
```

Do-while loop

```
#include<stdio.h>
void main()
{
    int num=1;
    do
    {
        if(num%2==0)
            printf("%d \t",num);
        num++;
    } while(num<=100);
}
```

For loop

```
#include<stdio.h>
void main()
{
    int num;
    for(num=1;num<=100;num++)
    {
        if(num%2==0)
            printf("%d \t",num);
    }
}
```

Output

2 4 6 8 10 12..... 100

Example4: find factorial of a number

Program

```
#include<stdio.h>
void main()
{
    int num,fact=1;

    printf("Enter a number:");
    scanf("%d",&num);

    while(num!=0)
    {
        fact=fact*num;
        num--;
    }
    printf("%d",fact);
}
```

Output

```
Enter a number: 5
120
```

Case Control Structure

- Switch
 - ‘switch’ keyword is used to design menu based programs.
 - Uses an integer constant or expression to evaluate.
 - ‘case’ keyword is used with a constant for different cases.
 - ‘default’ statements are executed by default.
 - ‘break’ keyword is used to break the flow of execution.

How the program is evaluated

- First switch is evaluated.
- The constant used in switch is compared with different cases.
- The constant which is found to be equal with the switch, the particular block of statement will be executed.
- The default statement will be executed by default independent of the cases.

Decision using switch

Syntax:

```
switch(integer expression)
{
    case const1:
        do this;
    case const2:
        do this;
    ....
    default:
        do this;
}
```

Example:

```
Switch(2)
{
    case 1:
        printf("One");
    case 2:
        printf("two");
    default:
        printf("Good One");
}
```

Example1: Understand the switch statement

Program

```
#include<stdio.h>
void main()
{
    int choice;

    printf("Enter your choice:");
    scanf("%d",&choice);

    switch(choice)
    {
        case 0:
            printf("First\n");
        case 1:
            printf("Second\n");
        case 2:
            printf("Third\n");
        default:
            printf("wrong choice");
    }
}
```

Output

```
Enter your choice:2
Second
Third
Wrong choice
```

Decision using switch

Syntax:

```
switch(integer expression)
{
    case const1:
        do this;
        break;
    case const2:
        do this;
        break;
    ....
    default:
        do this;
}
```

Example:

```
Switch(2)
{
    case 1:
        printf("One");
        break;
    case 2:
        printf("two");
        break;
    default:
        printf("Other than 1 or 2");
}
```

Example1: Digit to Word conversion

Program

```
#include<stdio.h>
void main()
{
    int num;
    printf("Enter a number between 0 to 9:");
    scanf("%d",&num);

    switch(num)
    {
        case 0:
            printf("Zero");
            break;
        case 1:
            printf("One");
            break;
        case 2:
            printf("Two");
            break;
        case 3:
            printf("Three");
            break;
        case 4:
            printf("Zero");
            break;
        case 5:
            printf("Five");
            break;
        case 6:
            printf("Six");
            break;
        case 7:
            printf("Seven");
            break;
        case 8:
            printf("Eight");
            break;
        case 9:
            printf("Nine");
            break;
        default:
            printf("Try again");
    }
}
```

Output

```
Enter a number between 0 to 9:5
five
```


Example 3: Arithmetic calculations on integer

```
/*P5.33 Program to perform arithmetic calculations on integers*/
#include<stdio.h>
main()
{
    char op;
    int a,b;
    printf("Enter number operator and another number : ");
    scanf("%d%c%d",&a,&op,&b);
    switch(op)
    {
        case '+':
            printf("Result = %d\n",a+b);
            break;
        case '-':
            printf("Result = %d\n",a-b);
            break;
        case '*':
            printf("Result = %d\n",a*b);
            break;
        case '/':
            printf("Result = %d\n",a/b);
            break;
        case '%':
            printf("Result = %d\n",a%b);
            break;
        default:
            printf("Enter valid operator\n");
    }/*End of switch*/
}/*End of main()*/
```

Output:

```
Enter number operator and another number : 2+5
Result = 7
```

Example 4: Vowel or consonant

```
/*P5.34 Program to find whether the alphabet is a vowel or consonant*/
#include<stdio.h>
main( )
{
    char ch;
    printf("Enter an alphabet : ");
    scanf("%c", &ch);
    switch(ch)
    {
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
            printf("Alphabet is a vowel\n");
            break;
        default:
            printf("Alphabet is a consonant\n");
    }
}
```