



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

Que.1) Explain the structure of a typical C Program. (W-18/6M) (S-16/7M) (W-15/4M) (W-14/7M)

Ans: Structure of a 'C' Program:

A 'C' Program may contain one or more section as shown below;

Documentation Section

Link Section

Definition Section

Global Declaration Section

main () Function Section

{

 Declaration Part

 Executable Part

}

Subprogram section

Function -1

Function – 2

(user – defined functions)

Function – n

Documentation section:

This section contains a set of comment line , which may contain author name, purpose of the program and such other details.

Link section:

This section contains the instructions which is to be provided to the compiler to link functions from the system library.

Definition Section:

This section defines all symbolic constants.

Global Declaration section:

Variables which are used in more than one functions are called global variables. These variables are declared in the global declaration section which is outside of all the functions.

This section declares all the user defined functions.

Every 'C' program must have one main () function section. This section contains two parts –

1) Declaration Part



Department of Electrical Engineering

Subject Notes – Computer Programming (Unit –I)

2) Executable Part

- The declaration part declares all the variables used in the executable part. There is atleast one statement in the executable part.
- These two parts must appear between the opening and closing braces. The program execution starts with the opening brace and ends with the closing brace.
- The closing brace of the main function section is the logical end of the program. All statements in the declaration and executable parts end with a semicolon (;). The subprogram section contains all the user defined functions that are called in the main function.
- User defined functions are generally placed immediately after the main function.

Que.2) Describe various data types supported by ‘C’. Give their ranges. (S-18/7M)(W-17/7M)(S-17/7M)(W-16/7M)(W-15/7M)(S-15/7M)

Ans: **Data Types:**

‘C’ supports three classes of data types:

- 1) Primary (or fundamental) data type.
- 2) Derived data types
- 3) User defined data types

Primary or fundamental data type:

All ‘C’ compilers supports five fundamental data types:

- (i) integer (int)
- (ii) character (char)
- (iii) floating point (float)
- (iv) double – precision floating point (double)
- (v) void

Primary Data Type

Integral Data Type		
Integer		Character
signed	unsigned type	
int	unsigned int	char
short int	unsigned short int	signed char
long int	unsigned long int	unsigned char
Floating Point type		
float	double long double	void



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

Size and range of basic data type on a 16 bit machine

Data Type	Range of Values
char	-128 to 127
int	-32768 to 32767
float	3.4 e -38 to 3.4 e +38
double	1.7 e -308 to 1.7 e +308

'C' has three classes of integer storage, namely

- i) short int
- ii) int
- iii) long int

in both signed and unsigned forms.

For a 16 bit machine, the range of unsigned integer numbers will be from 0 to 65, 535.

Long and unsigned integers is used to increase the range of values.

Floating Point Types:

Floating point numbers are defined in 'C' by the keyword float.

When the accuracy provided by a float number is not sufficient, the type double can be used to define the number. A double data type uses 64 bits, long double uses 80 bits

Void Types: The void type has no values. This is usually used to specify the type of functions. The type of functions is said to be void when it does not return any value the calling function. It can also play the role of a generic type, meaning that it can represent any of the other standard type.

Character Type:

- A single character can be defined as a character (char) type data.
- Characters are stored in 8 bits (one byte) of internal storage.
- The qualifier signed or unsigned may be explicitly applied to char.
- Unsigned chars have values between 0 to 255.
- Signed chars have values from -128 to 127.

Que.3) Define the terms variables and constants. (W-15/7M) (S-15/7M)

Ans: Variables:



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

A variable is a data name that is used to store a data value.

A variable can take different values at different times during execution.

Variable name may consists of letters, digits, and the underscore(_) character, subject to the following conditions:

- 1) Variables must begin with a letter.
- 2) Only the first eight characters are considered as most significant by many compilers.
- 3) Uppercase and lowercase are significant.
- 4) It should not be a keyword.
- 5) White space is not allowed.

Que.4) Define identifiers and keywords. (W-15/4M) (S-16/7M)

OR Discuss in brief keywords and identifiers. (S-15/7M)

And: **Keywords:**

Every 'C' word is classified as either a keyword or an identifier. All keywords have fixed meanings and these meanings cannot be changed.

Keywords are the basic building block for program statements.

Example: double, char, float, if, else, break etc.

All the keywords should start with (or must be written) in lowercase.

Identifiers:

- Identifiers refer to the names of variables, functions and arrays.
- These are user-defined and consists of a sequence of letters and digits, with a letter as a first character.
- Both uppercase & lowercase letters are permitted, although lowercase letters are commonly permitted or used.
- The underscore character is also permitted in identifiers. It is usually used as a link between two words in long identifiers.

Que. 5) What are the rules for constructing identifiers ?

(S-15/7M)

Ans: **Rules for identifiers:**

- i) First character must be an alphabet (or underscore)
- ii) Must consists of only letters, digits or underscore.
- iii) Only first 31 characters are significant.
- iv) Cannot use a keyword.



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

v) Must not contain whitespace.

Que.6) Explain different operators available in ‘C’.

(S-18/7M)

Ans: Operators:

C operators can be classified into a number of categories. They include

- i) Arithmetic Operators
- ii) Relational Operators
- iii) Logical Operators
- iv) Assignment Operators
- v) Increment & Decrement Operators
- vi) Conditional Operators
- vii) Bitwise Operators
- viii) Special Operators

Arithmetic Operators:

Operator	Meaning
+	Addition or unary plus
-	Subtraction or unary minus
*	Multiplication
/	Division
%	Modulo Division

Relational Operators:

Operator	Meaning
<	is less than
<=	is less than or equal to
>	is greater than
>=	is greater than or equal to
==	is equal to
!=	is not equal to

Logical Operators:

In addition to the relational operators, C has the following three logical operators

- && meaning logical AND
- || meaning logical OR
- ! meaning logical NOT

The logical operators && and || are used when we want to test more than one condition and make decisions.



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

Assignment Operators:

Assignment operators are used to assign the result of an expression to a variable. C has a set of 'shorthand' assignment operators of the form

$$V \text{ op} = \text{exp};$$

Where **v** is a variable, **exp** is an expression and **op** is a C binary arithmetic operator. The operator **op =** is known as the shorthand assignment operator.

Increment and Decrement Operators:

C allows two very useful operators not generally found in other languages. These are increment and decrement operators. (++ and --)

The operator ++ adds 1 to the operand and -- subtracts 1.

Both are unary operators and takes the following form:

$$++ m ; \text{ or } m ++ ; \text{ and } -- m ; \text{ or } m --$$

++ m ; is equivalent to m = m + 1 or (m+ =1;)

-- m ; is equivalent to m = m - 1 or (m- = 1;)

Increment and decrement operators are extensively used in for and while loops.

Conditional Operators:

A ternary operator pair “ ? : ” is available in C to construct conditional expressions of the form
exp1 ? exp 2; exp 3

where exp1, exp2, exp3 are expressions.

The operator ? : works as follows : exp1 is evaluated first. If it is nonzero (true), then the expression 2 is evaluated and becomes the expression value.

If exp1 is false. exp3 is evaluated & its value becomes the value of the expression.

Bitwise Operators:

Operator	Meaning
&	bitwise AND
	bitwise OR
^	bitwise exclusive OR
<<	shift left
>>	shift right

Special Operators:



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

C supports some special operators such as

- i) Comma operator
- ii) Size of operator
- iii) Pointer operator (& and *)
- iv) Member Selection operators (. and ->)

Que.7) How is input output carried out in 'C'?

(W-17/7M)

Ans: Input Output:

Reading a single character is done by using using function getchar.

Syntax:

```
variable_name = getchar ( );
```

variable_name is a valid C name that has been declared as char type.

Example:

```
char name;  
Name = getchar ( );
```

Formatted Input:

In C, for reading the formatted data, scanf function is used.

Syntax:

```
scanf ( " Control string ", arg1, arg2, ..... arg n);
```

The control string specifies the field format in which the data is to be entered arg1, arg2, arg n specify the address of locations where data is stored.

Control strings and arguments are separated by commas.

Control string contains field specifications which directs interpretations of input data. It may include:

Field (or format) specifications, consisting of the conversion character %, a data type character (or type specifier) , and an optional number, specifying the field width, Blanks, tabs or newlines.

Inputting Character Strings:

A scanf function can input strings containing more than one character. Following are the specifications for reading character strings.

% ws or %wc

Commonly used scanf format codes:



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

Code	Meaning
% c	Read a single character
% d	Read a decimal integer
% e	Read a floating point value
% f	Read a floating point value
% g	Read a floating point value
% h	Read a short integer
% i	Read a decimal, hexadecimal or octal integer
% o	Read an octal integer
% s	Read a string
% u	Read an unsigned decimal integer
% x	Read a hexadecimal integer
% [..]	Read a string of word (s)

Rules of scanf:

- 1) Each variable to be read must have a field specification.
- 2) For each field specification, there must be a variable address of proper type.
- 3) Any non- whitespace character used in the format string must have a matching character in the user input.
- 4) Never end the format string with whitespace. It is fatal error!
- 5) The scanf reads until :
 - A whitespace character is found in a numeric specifications or
 - The maximum number of character have been read or
 - An error is detected, or
 - The end of file is reached

Formatted Output:

In C printf function is used to print captions and numerical results.
The general form of printf statement is:

```
printf (“Control Strings”, arg1, arg2,....., argn);
```

printf never supplies a newline automatically and therefore multiple printf statements may be used to build one line of output.

A newline can be introduced by the help of newline character ‘\n’

Commonly used printf Format Codes

Code	Meaning
% c	Print a single character
% d	Print a decimal integer
% e	Print a floating point value in exponent form
% f	Print a floating point without exponent
% g	Print a floating point either e-type or f-type
% i	Print a signal decimal integer



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

% o	Print an octal integer, without leading zero
% s	Print a string
% u	Print an unsigned decimal integer
% x	Print a hexadecimal integer, without 0x

Commonly used Output Format Flags:

Flag	Meaning
-	Output is left – justified
+	+ or – will precede the signed numeric item
0	Causes leading zeros to appear
#	Causes octal and hex items to be preceded

Que.8) Give the syntax of the following control statements with example ?

- (i) **if statement** (ii) **if-else** (iii) **nested if-else** (iv) **else –if ladder**

Ans: C – language possesses such decision making capabilities by supporting the following statements:

1) if statement:

The if statement may be implemented in different forms depending on the complexity of conditions to be tested. The different forms are:

- i) simple if statement
- ii) if-else statement
- iii) nested if...else statement
- iv) else if ladder

Syntax of if statement:

```
if ( test expression)
{
    statement –block;
}
statement –x;
```

Syntax of if-else statement:

```
if ( test expression)
{
    True-Block statement(s);
}
else
{
    False-block statement (s);
}
Statement –x
```



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

Nested if-else statements:

When a series of decisions are involved, we may have to use more than one if Else statement in nested form as shown below;

```
if (test condition -1)
{
    if ( test condition -2 )
    {
        Statement – 1;
    }
    else
    {
        Statement -2;
    }
}
Statement -3;
}
Statement –x;
```

The else-if ladder:

When multipath decisions are involved, there is an another way of putting if's in which the statement associated with each **else** is an **if**.

Syntax:

```
if ( condition 1)
    Statement -1;
else if ( condition 2)
    Statement -2;
else if ( condition 3)
    Statement -3;
else if ( condition n)
    Statement –n;
else
    Default –statement;
Statement –x;
```

Que.9) Explain the various loop control statements in 'C' with their syntax. (S-17/7M)

Ans: Loop Control Statements:

In looping, a sequence of statements are executed until some conditions for termination of the loop are satisfied.

A looping process in general would include following four steps:



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

- 1) Setting and initialization of a condition variable.
- 2) Execution of statements in the loop.
- 3) Test for a specified value of the condition variable for execution of the loop.
- 4) Incrementing or updating the condition variable.

The 'C' language provides for three constructs for performing loop operations. They are:

- i) The while statement
- ii) The do statement
- iii) The for statement

while Statement:

Syntax:

```
while (test condition)
{
    body of the loop
}
```

- The while is an entry – controlled loop statement.
- The test condition is evaluated and if the condition is evaluated and if the condition is true, then the body of the loop is executed. After execution of the body, the test condition is once again evaluated and if it is true, the body is executed once again.
- This process is repeated until the test-condition finally becomes false and the control is transferred out of the loop.
- On exit, the program continues with the statement immediately after the body of the loop.
- The body of the loop may have one or more statements.

The do-while statement:

The while loop, makes the test condition before the loop is executed. Hence, if the condition is not satisfied at all, the body of the loop may not be executed at all. Such situations is handled by do statement.

Syntax:

```
do
{
    body of the loop
}
while ( test condition)
```



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

- The program proceeds to evaluate the body of the loop first.
- At the end of the loop, the test condition in while statement is evaluated.
- If the condition is true, the program continues to evaluate the body of the loop once again.
- This process continues as long as the condition is true
- When the condition becomes false, the loop will be terminated and the control goes to statement that appears immediately after the while statement.
- As the test condition is evaluated at the bottom of the loop, the do-while construct provides an exit-control loop and hence the body of the loop is always executed at least once.

The for statement:

The for loop is an entry controlled loop. It provides a more concise loop control structure.

Syntax:

```
for ( initialization;test-condition;increment)
{
body of the loop
}
```

The execution of the for statement is as follows:

1. Initialization of the control variables is done first, using assignment statements such as $i=1$ and $count = 0$. The variables i and $count$ are known as the loop control variables.
2. The value of the control variable is tested using the test-conditions. The test conditions is a relational expression the determines when the loop will exit. If the condition is true, the body of the loop is executed or else the loop is terminated and execution continues with the statement that immediately follows the loop.
3. When the body of the loop is executed, the control is transferred back to the for statement after evaluating the last statement in the loop. Now, the control variable is incremented using an assignment statement such as $i = i + 1$ and the new value of the control variable is again tested to see whether it satisfies the loop condition. This process continues till the value of the control variable fails to satisfy the test condition.



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

One important point about the for loop is that all the three actions namely initialization, testing and incrementing are placed in the for statement itself, making them visible to the programmers and users in one place.

Que. 10) Explain the differences between while and do-while loops with examples. (W-15/7M)

Ans:

Note: (the table below gives the comparison of the for, while and do-while loops. Students have to write the comparison which is asked in the question)

	for	while	do-while
1.	Entry Controlled	Entry Controlled	Exit Controlled
2	Initialization, testing and incrementing takes place in the for statement itself.	Testing is done in the while statement and if it is true then only increment is done in the statement.	Increment is done first and then the while condition is evaluated.
3	Loop is executed once the condition is satisfied in the for statement.	If the condition is not satisfied at very first attempt, loop may not execute at all.	Loop is executed first and then the condition is checked for the loop.
4	for (n=1; n<=10;n++) { _____ _____ } 	n=1; while (n<=10) { _____ _____ n++; } 	n=1; do { _____ _____ n++; } while (n< = 10);

Que. 11) Give the syntax of the following control statements with example.

i) do-while ii) if-else iii) while loop **(W-18/6M)(S-16/7M)**

Ans:

i) do-while loop:

Syntax:

```
do
{
body of the loop
}
while ( test – condition);
```



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

Example:

```
n=1;
do
{
-----
-----
n + +;
}
while ( n < = 10);
```

ii) If- else:

Syntax:

```
if ( test expression)
{
    True – block statement (s)
}
else
{
    False – block statement (s)
}
Statement – x;
```

Example:

```
if (a > b)
{
    printf ( “A is greater”);
}
else
{
    printf ( “B id greater”);
}
```

iii) while Loop:

Syntax:

```
while ( test condition)
{
    body of the loop
}
```

Example:

```
n=1;
while ( n< = 10)
{
-----
-----
n++;
}
```



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

Que.12) Describe various storage class in 'C' with examples. (W-17/7M)(S-17/7M)(W-16/7M)(S-16/7M)(S-15/7M)(W-14/7M)(S-14/7M)

Ans: Storage Class: Variables in C can have not only data type but also storage class that provides information about their location and visibility. The storage class decides the portion of the program within which the variables are recognized.

There are four storage classes in C:

- (a) Automatic storage class
- (b) Register storage class
- (c) Static storage class
- (d) External storage class

Automatic Storage Class:

The features of a variable defined to have an automatic storage class are as under:

Storage	- Memory
Default initial value	- Garbage value
Scope	- Local to the block in which the variable is defined
Life	- Till the control remains within the block in which the variable is defined

Register Storage Class:

The features to be defined as a register storage class are as under:

Storage	- CPU Variables
Default initial value	- Garbage value
Scope	- Local to the block in which the variable is defined
Life	- Till the control remains within the block in which the variable is defined

A value stored in a CPU register can always be accessed faster than the one that is stored in memory.

Hence if a variable is used in a program at many places, it is better to declare its storage class as register.

Static Storage Class:

The features of a variable defined to have a static storage class are as under:

Storage	- Memory
Default initial value	- Zero
Scope	- Local to the block in which the variable is defined
Life	- Value of the variable persists between different function calls.



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

External Storage Class:

The features of a variable whose storage class has to be defined as external are as follows:

Storage	- Memory
Default initial value	- Zero
Scope	- Global
Life	- As long as the program execution does'nt come to an end

External variables are declared outside to all functions that care to use them.

Que.13) Write a program in C to add the digits of a four digit number. Number is entered by user. (S-18/6M)(W-16/7M)

Ans:

```
#include <stdio.h>
main( )
{
    int num,a,n;
    int sum = 0;
    printf ( “ Enter a four digit number:\n”);
    scanf ( “%d”, & num);
    a = num % 10;
    n = num /10;
    sum = sum + a;
    a = n % 10;
    n = n / 10;
    sum = sum + a;
    a = n % 10;
    n = n / 10;
    sum = sum + a;
    a = n % 10;
    sum = sum + a;
    printf ( “ The sum of the four digit number is % d”, sum);
    getch( );
}
```

Note: (if in the question the number of digits is increased to five digit, the step $a = n \% 10$; $n = n / 10$; $sum = sum + a$; will be added one more time and if the digit is reduced to three digits, the step will be reduced by one time)

Que.14) A five digit number is entered through keyboard. Write a program in C to obtain the reverse number.

Ans:



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

```
# include <stdio.h>
main()
{
    int n, a, num ;
    int revnum = 0;
    printf ( “ Enter a five digit number:\n”);
    scanf ( “%d”, & num);
    a = num % 10;
    n = num /10;
    revnum = revnum + a* 10000;
    a = n % 10;
    n = n / 10;
    revnum = revnum + a * 1000;
    a = n % 10;
    n = n / 10;
    revnum = revnum + a * 100;
    a = n % 10;
    n = n / 10;
    revnum = revnum + a * 10;
    a = n % 10;
    revnum = revnum + a;
    printf ( “ The reverse of the five digit number is % d”, revnum);
    getch( );
}
```

Que.16) Write a program in ‘C’ to calculate net salary using formula:

Gross Salary = Basic + DA + HRA

Net Salary = Gross Salary – PT

Using Condition:

If Basic >= 2500 , PT =150 , HRA = 1200 and DA = 50 % of Basic

If Basic < 2500, PT =100 , HRA = 10 % of Basic and DA = 25 % of Basic (S-18/7M)

Ans:

```
# include <stdio.h>
main()
{
    float bs,da,hra,gs,ns;
    printf ( “Enter the Basic Salary “);
    scanf (“%f”, & bs);
    if (bs >= 2500)
    {
        pt = 150;
        hra = 1200;
        da = bs * 0.5;
    }
    else
    {
```



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

```
pt = 100;
hra = bs * 0.1;
da = bs * 0.25;
}
gs = bs + hra + da;
ns = gs - pt;
printf ( “ The Net Salary is %f”, ns);
getch( )
}
```

Que.17) In a company an employee is paid as under. If his basic salary is less than Rs. 1500/- then HRA = 10% of basic salary and DA = 90% of basic salary. If his basic salary is either equal to or above Rs. 1500/- then HRA = Rs. 500 and DA = 98 % of basic salary. If the basic salary is input through the keyboard. Write a program in C to find his gross salary. (W-18/7M)

Ans:

```
# include <stdio.h>
main( )
{
float bs,da,hra,gs,ns;
printf ( “Enter the Basic Salary “);
scanf (“%f”, & bs);
if (bs > 1500)
{
hra = bs * 0.1;
da = bs * 0.9;
}
else
{
hra = 500;
da = bs * 0.98;
}
gs = bs + hra;
printf ( “ The Gross Salary is %f”, gs);
getch( );
}
```

Que.18) Write a program to print the factorial of a number. The number is entered by the user. (W-17/7M)

Ans:

```
# include <stdio.h>
main ( )
{
int num,i = 1;
long int fact = 1;
printf ( “ Enter a number less than 34:\n”);
```



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

```
scanf ("%d", & num);
while ( i <= num)
{
    fact = fact * i);
    i ++;
}
printf (" The factorial of the entered number is %d",fact);
getch ();
}
```

Que. 19) Write a program to determine and print the sum of the following harmonic series for a given value of n: $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$. The value of 'n' is to be entered through keyboard. (W-16/7M)

Ans:

```
# include < stdio.h >
main ( )
{
    int i , n ;
    float sum = 0.0;
    printf ( " Enter the harmonic series number:\n");
    scanf ( " % d", & n);
    for ( i = 1; i <= n ; i ++ )
    {
        sum = sum + 1 / n;
    }
    printf ( " The sum of the harmonic series is %f", sum);
    getch ( );
}
```

Que.20) Write a function program to calculate factorial of a number and use it to find the sum of the series $- 1 + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{10!}$ (S-17/7M)

Ans:

```
# include < stdio.h >
main ( )
{
    int i=1 , j ;
    float fact, sum = 0.0;
    while ( i <= 10)
    {
        fact = 1.0;
        for ( j = 1; j <= i ; j ++ )
        {
            fact = fact * j;
        }
    }
}
```



Department of Electrical Engineering
Subject Notes – Computer Programming (Unit –I)

```
sum = sum + i / fact;
    }
    i ++;
}
printf ( “ The sum of series is %f “, sum);
getch ( );
}
```