



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)  
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu.



## MUST KNOW CONCEPTS

MKC

IT

2020-21

Course Code & Course Name : 19GES01 & Programming For Problem Solving Using C  
Year/Sem/Sec : I / I / -

S.No.	Term	Notation (Symbol)	Concept / Definition / Meaning / Units / Equation / Expression	Units
<b>Unit-I : Introduction to C Programming</b>				
1.	Computer		A programmable electronic device designed to accept data, perform prescribed mathematical and logical operations at high speed	
2.	Components of Computer System		<ul style="list-style-type: none"> <li>• Hardware</li> <li>• Software</li> </ul>	
3.	Hardware		Electronic device, hence one can touch and see the hardware	
4.	Computer Hardware categories		<ul style="list-style-type: none"> <li>• Input Devices</li> <li>• Output Devices</li> <li>• Secondary Storage Devices</li> <li>• CPU</li> </ul>	
5.	Software		Software is a set of programs, which is designed to perform a well-defined function	
6.	Software categories		<ul style="list-style-type: none"> <li>• Application Software</li> <li>• System Software</li> </ul>	
7.	System Software		Collection of programs designed to operate, control, and support the operation of computer itself <b>Ex :</b> compiler, assembler, debugger, driver, etc	
8.	Application Software		Program that accomplish user task. <b>Ex:</b> Word processor, Railways Reservation S/W, etc	
9.	High-Level Language		Easy to understand and human-readable program <b>Ex:</b> C++, C, JAVA, FORTRAN, etc	
10.	Machine Language		Binary digits 0 and 1, understood and read by a computer system	
11.	Program Design Tools		Tools used to develop a program. Ex: Algorithms, Flowcharts, Pseudo codes	
12.	Algorithms		Sequence of steps to solve a particular problem	
13.	Flowcharts		Diagram that represents a workflow or process	

14.	Pseudo codes		False code represents the program, It cannot be compiled or run like a regular program	
15.	main()		Main function from where the program execution begins	
16.	scanf()		Commonly used function to take input from the user	
17.	printf()		This function is used for displaying the output on the screen	
18.	Keyword		Predefined, reserved words used in programming, It have a special meaning. <b>Ex:</b> int, switch, for ect...	
19.	Identifier		Name given to entities such as variables, functions, structures <b>Ex:</b> int a, b; / Here, a and b are identifiers	
20.	Rules for naming identifiers		The first letter of an identifier should be either a letter or an underscore, it can be followed by letter/digit	
21.	Data Type		Specifies the type of data that a variable can store such as integer <b>Ex:</b> Basic DT in C are Basic data types	
22.	Constants		Fixed values that the program may not alter during its execution	
23.	Variables		Name of the memory location, used to store data and its value can be changed, and it can be reused many times	
24.	Operators in C		Operators to perform tasks including arithmetic, conditional and bitwise operations	
25.	Typecasting		Type casting is a way to convert a variable from one data type to another data type	

### Unit-II : Conditional and Looping Statements

26.	Conditional branching Statements		Conditional branching Statements help to jump from one part to the program to another depending on whether a particular condition is satisfied or not	
27.	Types of conditional statements in C		<ul style="list-style-type: none"> <li>• if statement</li> <li>• if-Else statement</li> <li>• if-Else if ladder</li> <li>• Switch statement</li> </ul>	
28.	if statement		if statement in C language is used to execute the code if a condition is true	
29.	if-else statement		Statement in C language is used to execute the code if condition is true or false	
30.	Nested if-else statement		if statement inside an if statement is known as nested if	
31.	If – else – if ladder Statement		Executes one condition from multiple statements. The execution starts from top and checked for each if condition	
32.	Switch Statement		Switch statement is an alternative to long if-else-if ladders. The expression is checked for different cases and the one match is executed	

33.	Break Statement		Break statement is used to move out of the switch	
34.	Looping Statement		Looping statement are the statements execute one or more statement repeatedly several number of times	
35.	Type of Loops		<ul style="list-style-type: none"> <li>• while loop</li> <li>• for loop</li> <li>• do..while</li> </ul>	
36.	Difference between conditional and looping statement		Conditional statement executes only once in the program where as looping statements executes repeatedly several number of time	
37.	Nested Loop		Nested loop one loop is place within another loop body	
38.	While Loop		Executes a statement or a block of statements until the specified boolean expression evaluates to false	
39.	For Loop		This loop has three sections - index declaration, condition (boolean expression) and incremented/ decremented section	
40.	Continue statement		Continue statement is inside a loop, control jumps to the beginning of the loop for next iteration, skipping the execution of statements inside the body of loop for the current iteration.	
41.	Statement		A statement is a command given to the computer that instructs the computer to take a specific action	
42.	Boolean values		<ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul>	
43.	Counter controlled loop		we know that exactly how many times loop body will be executed	
44.	Sentinel controlled loop		we don't know about the loop recurrence, Execution of loop is based on condition not counter.	
45.	Is nested loop possible in C?		Yes, it is possible. We can use loop with in the loop any number of times.	
46.	Fall down property		In switch statement, break is not used after the block statements, then it will execute next case or default statements until break not reached or switch not finished.	
47.	Which loop statement is executed at least once even loop test condition if false?		do while loop executes once even loop test condition if false	
48.	Jumping statements in C		<ol style="list-style-type: none"> <li>1. goto</li> <li>2. break</li> <li>3. continue</li> </ol>	
49.	Infinite loop		A loop which is never finished is known as infinite loop	
50.	Function pointer		When a pointer points to a function, it is known as function pointer or pointer to a function.	

### Unit-III : Functions and Arrays

51.	Functions		A function is a self-contained block or a sub-program of one or more statements that performs a special task when called.	
52.	Function Declaration		Function declaration is a declaration statement that identifies a function with its name, a list of arguments that it accepts and the type of data it returns.	
53.	Void Function		Return statement with no returned value.	
54.	Function Call		A function call is a request made by a program that performs a predetermined function	
55.	Call By Value		Call by value in which values of the variables are passed by the calling function to the called function.	
56.	Call By Reference		Call by reference in which address of the variables are passed by the calling function to the called function.	
57.	Arrays		An array is a collection of similar data elements. The elements of the array are stored in consecutive memory locations and are referenced by an index (also known as the subscript).	
58.	Linear Search		Linear search is also called sequential search. Linear search is a method for searching a value within a array.	
59.	Binary Search		Binary search works on sorted arrays. Binary search begins by comparing an element in the middle of the array with the target value.	
60.	Two Dimension(2D-Array)		A two dimensional array is specified using two subscripts where one subscript denotes row and the other denotes column.	
61.	int main		int main means that our function needs to return some integer at the end of the execution and we do so by returning 0 at the end of the program.	
62.	Types of function		<ul style="list-style-type: none"> <li>• Predefined functions</li> <li>• User defined functions</li> </ul>	
63.	How to read the Matrix		<pre>for (c = 0; c &lt; m; c++)     for (d = 0 ; d &lt; n; d++)         scanf("%d", &amp;second[c][d]);</pre>	
64.	Formula for Addition of two matrix		sum[c][d] = first[c][d] + second[c][d];	
65.	\n		New Line	
66.	scanf( )		scanf( ) allows to read more than just a single character at a time.	
67.	Why header files are included in 'C' programming?		Each header file has 'h' extension and include using '# include' directive at the beginning of a program.	
68.	Define delimiters in 'C'.		: ; ( ) [] {} # ,	

69.	What is meant by Recursive function?		If a function calls itself again and again, then that function is called Recursive function.						
70.	Is it possible to place a return statement anywhere in 'C' program?		Yes. The return statement can occur anywhere.						
71.	types of errors occurred in C program		1. Syntax errors 2. Runtime errors 3. Logical errors 4. Latent errors						
72.	What are the types of Arrays?		1. One-Dimensional Array 2. Two-Dimensional Array 3. Multi-Dimensional Array						
73.	Typedef		It is used to create a new data using the existing type. <b>Syntax:</b> typedef data type name;						
74.	Operator overloading		Operator overloading is a compile-time polymorphism in which the operator is overloaded to provide the special meaning to the user-defined data type						
75.	Function overriding		Function overriding is a feature that allows us to have a same function in child class which is already present in the parent class.						
<b>Unit-IV : Strings and Structures</b>									
76.	String		Sequence of characters						
77.	String Functions in C		1 strcpy(s1,s2); 2 strcat(s1, s2); 3 strlen(s1); 4 strcmp(s1,s2); 5 strchr(s1,ch); 6 strstr(s1,s2);						
78.	Typedef		It is used to create a new data using the existing type. <b>Ex:</b> typedef int hours: hours hrs; /* Now, hours can be used as new datatype */						
79.	Types of sorting		<ul style="list-style-type: none"> <li>• Insertion sort.</li> <li>• Merge Sort.</li> <li>• Heap Sort</li> <li>• Selection sort</li> <li>• Bubble sort</li> <li>• Quick Sort.</li> <li>• Radix Sort.</li> </ul>						
80.	How to find the length of a string		Given a string str. Using function strlen(Str)						
81.	Converting String Into Uppercase		Convert a String to Uppercase in C using functionstrupr ()						
82.	Converting string into lower case		Convert a String to Lowercase in C using function strlwr(chars)						
83.	Declare a string		char s[5]; s[0] s[1] s[2] s[3] s[4] <table border="1" style="margin-left: 20px;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>						

84.	Initialize strings		<pre>char c[] = "abcd"; char c[50] = "abcd";</pre>	
85.	Passing Strings to Functions		Strings can be passed to a function in a similar way as arrays.	
86.	Strings and Pointers		Similar like arrays, string names are "decayed" to pointers. Hence, you can use pointers to manipulate elements of the string.	
87.	Structure		Use to combine data of different types together	
88.	Declaring Structure Variables		<pre>struct structure _ name { structure_element 1; structure_element 2; structure_element 3; }; struct structure_name v1,v2...vn; v1,v2....vn are structure variable</pre>	
89.	Declaring structure using pointer variable		<pre>struct student *report, rep;</pre>	
90.	Accessing structure members using normal variable		<pre>report.mark; report.name; report.average;</pre>	
91.	Accessing structure members using pointer variable		<pre>eport -&gt; mark; report -&gt; name; report -&gt; average;</pre>	
92.	Rules for declaring a structure		The template is terminated with a semicolon The tag name such as book _ bank can be used to declare structure variables of its type, later in the program.	
93.	Accessing structure elements		After declaring the structure type, variables and members, the member of the structure can be accessed by using the structure variable along with the dot(.) operator.	
94.	Access pointer members of a structure		-> - Structure pointer operator	
95.	Structure Initialization		Like any other data type, a structure variable can be initialized at compile time. <pre>main() { struct { int weight; float height; } student ={60, 180.75};</pre>	
96.	Nested structures		Structure with in another structure is called nested structure	
97.	Copying and Comparing Structure Variable		Two variables of the same structure type can be copied the same way as ordinary variables. If e1 and e2 belong to the same type, then the following statement is valid. e1 = e2, and e2 = e1;	

98.	Structure Can Be Accessed In two Ways In a C Program		<ul style="list-style-type: none"> <li>Using normal structure variable</li> <li>Using pointer variable</li> </ul>	
99.	Pointers to Structures		struct Books *struct_pointer;	
100.	Structure Padding		In order to align the data in memory one or more empty bytes (addresses) are inserted (or left empty) between memory addresses which are allocated for other structure members while memory allocation	

### Unit-V : Pointers and File Processing

101.	Pointers		A pointer is a variable whose value is the address of another variable	
102.	Declaring a pointer		Datatype * pointer-name;	
103.	Accessing a Variable through its Pointer		The indirection operator (*) is used to access the value of a variable by its ptr * can be remembered as value at address	
104.	Null pointer		A pointer is said to be null pointer when its right value is 0. A null pointer can never point to valid data.	
105.	Pointer to pointer		chain of pointers. int **var;	
106.	& var		Address of var variable	
107.	*var		Value of *ip variable	
108.	FILE		File is a collection of bytes that is stored on secondary storage devices like disk	
109.	Opening/Creating a file		fopen() – To open a file FILE *fopen (const char *filename, const char *mode)	
110.	Closing a file		fclose() – To close a file <b>Declaration:</b> int fclose(FILE *fp);	
111.	Reading a file		fgets() – To read a filechar *fgets(char *string, int n, FILE *fp)	
112.	Writing in a file		fprintf() – To write into a file <b>Declaration:</b> int fprintf(FILE *fp, const char *format);	
113.	File Processing		A file represents a sequence of bytes, regardless of it being a text file or a binary file.	
114.	Opening Files		fopen( ) function is used to create a new file or to open an existing file	
115.	fseek ( )		fseek ( ) function moves file pointer position to given location.	
116.	SEEK_SET		SEEK_SET moves file pointer position to the beginning of the file	
117.	SEEK_CUR		SEEK_CUR moves file pointer position to the beginning of the file.	
118.	SEEK_END		SEEK_END moves file pointer position to the end of file.	
119.	ftell ( )		ftell ( ) function gives current position of file pointer	

120.	rewind ()		rewind () function moves file pointer position to the beginning of the file.	
121.	remove ()		remove () function deletes a file.	
122.	fflush ()		fflush () function flushes a file.	
123.	File mode		r- Opens an existing text file for reading purpose w- Opens a text file for writing <b>a-</b> appending mode r+ / w+ - Opens file for both reading and writing	
124.	Sequential access file		A sequential access file is such that data are saved in sequential order: one data is placed into the file after another	
125.	Random access file		If the amount of data stored in a file is fairly large, the use of random access will allow you to search through it quicker.	
<b>Placement Questions</b>				
126.	iterator protocol		iter()- To create an iterator next()- To iterate to the next element	
127.	Tuple packing		we place value into a new tuple	
128.	Tuple unpacking		we extract those values back into variables.	
129.	frozen set		Frozen set is immutable, we cannot change its values.	
130.	Dogpile effect		In case the cache expires, what happens when a client hits a website with multiple requests is what we call the dogpile effect.	
131.	JSON		JSON stands for JavaScript Object Notation.	
132.	Garbage collection		form of automatic memory management which attempts to reclaim no longer use of memory	
133.	sub()		This looks for all substrings where the regex pattern matches, and replaces them with a different string	
134.	subn()		Like sub(), this returns the new string and the number of replacements made	
135.	map()		This function applies a function to each element in the iterable.	
136.	filter()		This function lets us keep the values that satisfy some conditional logic.	
137.	reduce()		This function reduces a sequence pair	
138.	Lamda()		A lambda function is a small anonymous function. It can take any number of arguments, but can only have one expression.	
139.	Is Python call-by-value or call-by-reference?		Python is neither call-by-value, nor call-by-reference. It is call-by-object-reference	



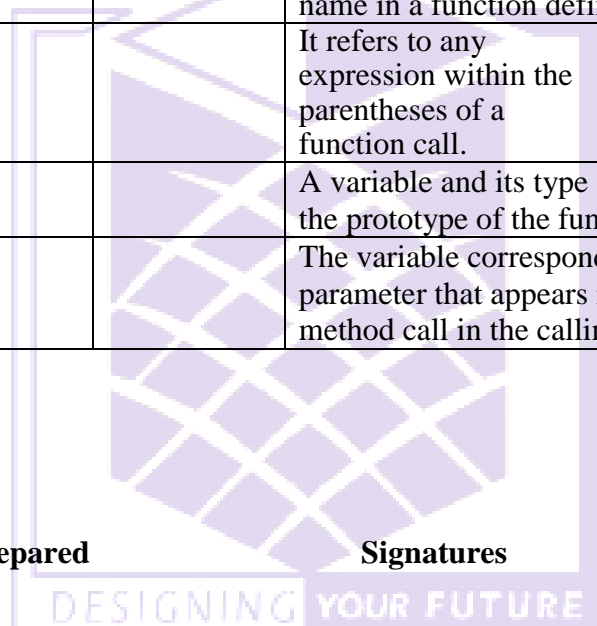
140.	<code>__init__()</code>		<code>__init__()</code> is what we need to initialize a class when we initiate it.	
141.	Case sensitive		Python is a case-sensitive language. This means, Variable and variable are not the same	
142.	Name famous python interpreter		Cpython & jpython	
143.	Extension of python file		PY is a script file format used by Python	
144.	Pointer		Variable that contains address of another variable	
145.	Structure		Structure is another user defined data type available in C that allows to combine data items of different kinds.	
146.	Union		A union is a special data type available in C that allows to store different data types in the same memory location.	
147.	Parameter		It refers to any declaration within the parentheses following the function name in a function definition;	
148.	Argument		It refers to any expression within the parentheses of a function call.	
149.	Formal Parameter		A variable and its type as they appear in the prototype of the function or method.	
150.	Actual Parameter		The variable corresponding to a formal parameter that appears in the function or method call in the calling environment.	

**Faculty Team Prepared**

**Signatures**

**HoD**

1. Mr.S.GOPI



**Estd. 2000**