|  |  |  |
| --- | --- | --- |
|  | **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** |
|  |  |  | | | | |  |
| **CSE** | |  | | | | | **2021-22** |
|  | | |  |  | | | |
| **Course Code & Course Name** | | | **:19GES02&Programming for Problem Solving Techniques** | | |  | | |

**Year/Sem/Sec :I /I / MECH**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Term** | **Notation**  **(Symbol)** | | **Concept / Definition / Meaning /**  **Units / Equation / Expression** | | **Units** |
| **Unit I :Introduction To C Programming** | | | | | | |
|  | Computer |  | | A programmable electronic device designed to accept data, perform prescribed mathematical and logical operations at high speed | |  |
|  | Components of Computer System |  | | Hardware and Software | |  |
|  | Hardware |  | | Electronic device, hence one can touch and see the hardware | |  |
|  | Computer Hardware categories |  | | Input Devices  Output Devices  Secondary Storage Devices  CPU | |  |
|  | Software |  | | Software is a set of programs, which is designed to perform a well-defined function | |  |
|  | Software categories |  | | Application Software  System Software | |  |
|  | System Software |  | | collection of programs designed to operate, control, and support the operation of computer itself  Ex :compiler, assembler, debugger, driver, etc | |  |
|  | Application Software |  | | Program that accomplish user task.  Ex:Word processor, Railways Reservation S/W ,etc | |  |
|  | High-Level Language |  | | Easy to understand and human-readable program  Examples: C++, C, JAVA, FORTRAN, etc | |  |
|  | Machine Language |  | | Binary digits 0 and 1, these binary digits are understood and read by a computer system | |  |
|  | Program Design Tools |  | | Tools used to develop a program. Ex: Algorithms, Flowcharts, Pseudo codes | |  |
|  | Algorithms |  | | Sequence of steps to solve a particular problem | |  |
|  | Flowcharts |  | | Diagram that represents a workflow or process | |  |
|  | Pseudo codes |  | | False code represents the program, It cannot be compiled or run like a regular program | |  |
|  | main() |  | | main function from where the program execution begins | |  |
|  | scanf() |  | | Commonly used function to take input from the user | |  |
|  | printf() |  | | This function is used for displaying the output on the screen | |  |
|  | Keyword |  | | Predefined, reserved words used in programming, It have a special meaning. Ex: [int](https://www.programiz.com/c-programming/list-all-keywords-c-language#int), [switch](https://www.programiz.com/c-programming/list-all-keywords-c-language#switch_case_default), [for](https://www.programiz.com/c-programming/list-all-keywords-c-language#for) ect… | |  |
|  | Identifier |  | | Name given to entities such as variables, functions, structures Ex: int a, b; / Here, a and b  are identifiers | |  |
|  | 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 | |  |
|  | Data Type |  | | Specifies the type of data that a variable can store such as integer Ex: Basic DT in C are Basic data types | |  |
|  | Constants |  | | Fixed values that the program may not alter during its execution | |  |
|  | Variables |  | | Name of the memory location, used to store data and its value can be changed, and it can be reused many times | |  |
|  | Operators in C |  | | Operators to perform tasks including arithmetic, conditional and bitwise operations | |  |
|  | Typecasting |  | | Type casting is a way to convert a variable from one data type to another data type | |  |
| Unit II : Conditional and Looping Statements | | | | | | |
|  | 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 | |  |
|  | Types of conditional statements in C |  | | if statement  if-Else statement  if-Else if ladder  Switch statement | |  |
|  | if statement |  | | if statement in C language is used to execute the code if a condition is true | |  |
|  | if-else statement |  | | Statement in C language is used to execute the code if condition is true or false | |  |
|  | Nested if-else statement |  | | if statement inside an if statement is known as nested if | |  |
|  | If – else – if ladder Statement |  | | Executes one condition from multiple statements. The execution starts from top and checked for each if condition | |  |
|  | 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 | |  |
|  | Break Statement |  | | Break statement is used to move out of the switch | |  |
|  | Looping Statement |  | | Looping statement are the statements execute one or more statement repeatedly several number of times | |  |
|  | Type of Loops |  | | while loop  for loop  do..while | |  |
|  | Difference between conditional and looping statement |  | | Conditional statement executes only once in the program where as looping statements executes repeatedly several number of time | |  |
|  | Nested Loop |  | | Nested loop one loop is place within another loop body | |  |
|  | While Loop |  | | Executes a statement or a block of statements until the specified boolean expression evaluates to false | |  |
|  | For Loop |  | | This loop has three sections - index declaration, condition (boolean expression) and incremented/ decremented section | |  |
|  | 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. | |  |
|  | Statement |  | | A statement is a command given to the computer that instructs the computer to take a specific action | |  |
|  | Boolean values |  | | True  False | |  |
|  | Counter controlled loop |  | | we know that exactly how many times loop body will be executed | |  |
|  | Sentinel controlled loop |  | | we don’t know about the loop recurrence, Execution of loop is based on condition not counter. | |  |
|  | Is nested loop possible in C? |  | | Yes, it is possible. We can use loop with in the loop any number of times. | |  |
|  | 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. | |  |
|  | 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 | |  |
|  | Jumping statements in C |  | | 1. goto 2. break 3. continue 4. return | |  |
|  | Infinite loop |  | | A loop which is never finished is known as infinite loop | |  |
|  | 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 | | | | | | |
|  | Functions |  | | A function is a self-contained block or a sub-program of one or more statements that performs a special task when called. | |  |
|  | 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. | |  |
|  | Void Function |  | | A function with void result type ends either by reaching the end of the function or by executing a return statement with no returned value. | |  |
|  | Function Call |  | | A function call is a request made by a program that performs a predetermined function | |  |
|  | Call By Value |  | | Call by value in which values of the variables are passed by the calling function to the called function. | |  |
|  | Call By Reference |  | | Call by reference in which address of the variables are passed by the calling function to the called function. | |  |
|  | 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). | |  |
|  | Linear Search |  | | Linear search is also called sequential search. Linear search is a method for searching a value within a array. | |  |
|  | 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. | |  |
|  | Two Dimension(2D-Array) |  | | A two dimensional array is specified using two subscripts where one subscript denotes row and the other denotes column. | |  |
|  | 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. | |  |
|  | Types of function |  | | Predefined functions  User defined functions | |  |
|  | How to read the Matrix |  | | for (c = 0; c < m; c++)  for (d = 0 ; d < n; d++)  scanf("%d", &second[c][d]); | |  |
|  | Formula for Addition of two matrix |  | | sum[c][d] = first[c][d] + second[c][d]; | |  |
|  | /n |  | | New Line | |  |
|  | scanf( ) |  | | scanf( ) allows to read more than just a single character at a time. | |  |
|  | 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. | |  |
|  | Define delimiters in ‘C’. |  | | : ; () [] {} # , | |  |
|  | What is meant by Recursive function? |  | | If a function calls itself again and again, then that function is called Recursive function. | |  |
|  | Is it possible to place a return statement anywhere in ‘C’ program? |  | | Yes. The return statement can occur anywhere. | |  |
|  | types of errors occurred in C program |  | | 1. Syntax errors  2. Runtime errors  3. Logical errors  4. Latent errors | |  |
|  | What are the types of Arrays? |  | | 1.One-Dimensional Array  2. Two-Dimensional Array  3. Multi-Dimensional Array | |  |
|  | typedef |  | | It is used to create a new data using the existing type. Syntax: typedef data type name; | |  |
|  | 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. | |  |
|  | 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. | |  |
| Unit IV: Introduction to Python Programming | | | | | | |
|  | Python |  | | Python is an interpreted, object-oriented, high-level programming language designed to be easy to read and simple to implement. | |  |
|  | Interpreter |  | | To execute a program in a high level languae by translating it one line at a time . | |  |
|  | Compiler |  | | To translate a program Written in High level Language into Low level Language all at once,inpreparation for later Execution. | |  |
|  | Interactive Mode |  | | Interactive mode is used when an user wants to run one single line or one block of code. It runs very quickly and gives the output instantly. | |  |
|  | Script Mode |  | | Script Mode is used when the user is working with more than one single code or a block of code and  where the scripted and finished .py files are run in the Python interpreter | |  |
|  | Expression |  | | An expression is a combination of values ,variables and operators. | |  |
|  | Fruitful Function |  | | A function that yields a return value instead of None. incremental development. A program development plan intended to simplify debugging by adding and testing only a small amount of code at a time. | |  |
|  | Pass Statement |  | | The pass statement is a null operation; nothing happens when it executes. | |  |
|  | Single line Comment |  | | #- hash symbol | |  |
|  | Recursive Function |  | | A recursive function can be defined as a routine that calls itself directly or indirectly. | |  |
|  | Advantages of Interactive mode |  | | Python, in interactive mode, is good enough to learn, experiment or explore.  * + Working in interactive mode is convenient for beginners and for testing small pieces of code. | |  |
|  | Disadvantages of Interactive mode |  | | * + We cannot save the statements and have to retype all the statements once again to re-run them.   + In interactive mode, you type Python programs and the interpreter displays the result: | |  |
|  | When to use Script mode |  | | If we are very clear about the code, then we use script mode | |  |
|  | Advantages of Script mode |  | | * Save the statements for future use * No need to retype all the statements and to rerun them | |  |
|  | Value |  | | Value can be any letter ,number or string. | |  |
|  | Numbers |  | | Numbers store numerical values | |  |
|  | Types of numbers supported by python |  | | Integer,Long,Float and Complex | |  |
|  | type-coercion |  | | Implicit conversion of datatypes | |  |
|  | Type-checking can be done at two stages |  | | Static and Dynamic | |  |
|  | Static |  | | Data Types are checked before execution | |  |
|  | Dynamic |  | | Data Types are checked during execution. | |  |
|  | Expression |  | | A sequence of operands and operators is called an expression like a + b - 5 , | |  |
|  | Exit() |  | | Scripts normally exit when the interpreter reaches the end of the file, but we may also call for the program to exit explicitly with the built-in exit functions. | |  |
|  | List other functions for exit() |  | | quit(),sys.exit() and os.exit() | |  |
|  | Types of conditional statements |  | | If-else  Elif  Switch | |  |
| Unit V- Strings, Lists, Tuples and Dictionaries | | | | | | |
|  | Slice() |  | | The slice() constructor creates a slice object representing the set of indices specified by range(start, stop, step). | |  |
|  | Sequence |  | | A sequence is an Ordered Collection of items,indexed by positive integers.It is a Combination of Mutable and Immutabledatatypes | |  |
|  | List |  | | List is an ordered sequence of items.It can be written as a list of comma-separated items (values) between square brackets[ ]. | |  |
|  | Mention any 5 list methods |  | | append() ,extend () ,sort(), pop(),index(),insert and remove() | |  |
|  | List Vs Dictionary |  | | List is a mutable type meaning that it can be modified, list can store a sequence of objects in a certain order whereas dictionary is immutable and is a key value store and is not ordered. | |  |
|  | aliasing |  | | An object with more than one reference has more than one name, then the object is said to be aliased. | |  |
|  | Cloning |  | | In order to modify a list and also keep a copy of the original, it is required to make a copy of the list itself, not just the reference. | |  |
|  | List parameters |  | | Passing a list as an argument actually passes a reference to the list, not a copy of the list. | |  |
|  | Tuple |  | | A tuple is same as list, except that the set of elements is enclosed inparentheses()instead of square brackets[].A tuple is an immutable list. | |  |
|  | Benefits of Tuple assignment |  | | It is often useful to swap the values of two variables. With conventional assignments a temporary variable would be used | |  |
|  | Indexing |  | | The index operator [] to access an item in a tuple where the index starts from 0.Ex: a tuple having 6 elements will have indices from 0 to 5. | |  |
|  | String |  | | A String in Python consists of a series or sequence of characters - letters, numbers, and special characters. Strings are immutable. | |  |
|  | Dictionaries |  | | Lists are ordered sets of objects, whereas dictionaries are unordered sets.Dictionary is created by using curly brackets. i,e. {} | |  |
|  | Key-value pairs |  | | The elements of a dictionary appear in a comma-separated list. Each entry contains an index and a value separated by a colon. In a dictionary, the indices are called keys, so the elements are called key-value pairs. | |  |
|  | Two basic dictionary operations |  | | Del -removes key-value pairs from a dictionary   Len - returns the number of key-value pairs | |  |
|  | Negative Indexing |  | | Python allows negative indexing for its sequences. The index of -1 refers to the last item, -2 to the second last item | |  |
|  | Quicksort |  | | QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. | |  |
|  | Merge Sort |  | | MergeSortis a Divide and Conquer algorithm. It divides input array in two halves, calls itself for the two halves and then merges the two sortedhalves. | |  |
|  | searching |  | | It refers to the process of finding location LOC of an element in a list. T | |  |
|  | sorting |  | | The arrangement of data in a preferred order is called sorting | |  |
|  | Benefits of Tuples |  | | Tuples are faster than lists.  * If the user wants to protect the data from accidental changes, tuple can be used. * Tuples can be used as keys in dictionaries, while lists can't. | |  |
|  | Concatenation |  | | Adding tuple elements at the end of another tuple element | |  |
|  | Repetition |  | | Creates new string, concatenating multiple copies of same string | |  |
|  | State |  | | Transition from one process to another process under specified condition with in a time is called state. | |  |
|  | Break and continue |  | | It terminates the current loop and executes the remaining statement outside the loop. It terminates the current iteration and transfer the control to the next iteration in the loop. | |  |
| Placement Questions | | | | | | |
|  | iterator protocol |  | | iter()- To create an iterator  next()- To iterate to the next element | |  |
|  | Tuple packing |  | | we place value into a new tuple | |  |
|  | Tuple unpacking |  | | we extract those values back into variables. | |  |
|  | frozen set |  | | Frozen set is immutable ,we cannot change its values. | |  |
|  | 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. | |  |
|  | JSON |  | | JSON stands for JavaScript Object Notation. | |  |
|  | Garbage collection |  | | form of automatic [memory management](https://en.wikipedia.org/wiki/Memory_management) which attempts to reclaim no longer use of memory | |  |
|  | sub() |  | | This looks for all substrings where the regex pattern matches, and replaces them with a different string | |  |
|  | subn() |  | | Like sub(), this returns the new string and the number of replacements made | |  |
|  | map() |  | | This function applies a function to each element in the iterable. | |  |
|  | filter() |  | | This function lets us keep the values that satisfy some conditional logic. | |  |
|  | reduce() |  | | This function reduces a sequence pair | |  |
|  | Lamda() |  | | A lambda function is a small anonymous function.  It can take any number of arguments, but can only have one expression. | |  |
|  | 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 | |  |
|  | \_\_init\_\_() |  | | \_\_init\_\_() is what we need to initialize a class when we initiate it. | |  |
|  | Case sensitive |  | | Python is a case-sensitive language. This means, Variable and variable are not the same | |  |
|  | Name famous python interpreter |  | | Cpython & jpython | |  |
|  | Extension of python file |  | | PY is a script file format used by Python | |  |
|  | pointer |  | | Variable that contains address of another variable | |  |
|  | Structure |  | | Structure is another user defined data type available in C that allows to combine data items of different kinds. | |  |
|  | Union |  | | A union is a special data type available in C that allows to store different data types in the same memory location. | |  |
|  | Parameter |  | | It refers to any declaration within the parentheses following the function name in a function definition; | |  |
|  | argument |  | | It refers to any expression within the parentheses of a function call. | |  |
|  | Formal Parameter |  | | A variable and its type as they appear in the prototype of the function or method. | |  |
|  | 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** | |  | |
|  | Mrs.T.Divya | |  | |
|  | Ms.R.Nivethitha | |  | |  | |

**HOD**