**UNIT I - BASICS OF C PROGRAMMING**

**PART-A**

1. Write about features and applications of C? **[CO1, K2]**

**Features in C**

* Portable
* Function rich library
* Case sensitive
* Robustness
* Modularity

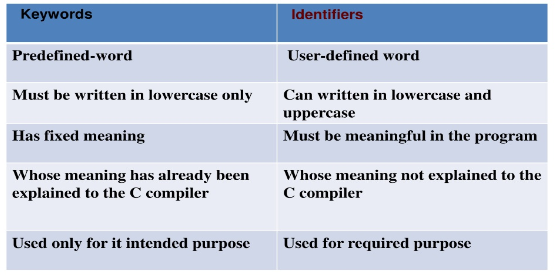
**Applications of C**

* Operating Systems
* GUI (Graphical User Interface)
* Embedded Systems
* Google
* Design of a Compiler
* Mozilla Firefox and Thunderbird

1. Define the rules for writing identifier names in C. **[CO1, K1]**

* An Identifier can only have alphanumeric characters (a-z, A-Z, 0-9) and underscore (\_).
* The first character of an identifier can only contain alphabet (a-z, A-Z) or underscore (\_).
* Identifiers are also case sensitive in C. For example name and Name are two different identifier.
* Keywords are not allowed to be used as Identifiers.
* No special characters, such as semicolon, white spaces, slash or comma are permitted to be used in Identifier.

1. Differentiate keyword and identifiers in C. **[CO1, K2]**

****

1. Enumerate Constants in C. **[CO1, K2]**

Constants are the terms that can't be changed during the execution of a program.

The types of constants are:

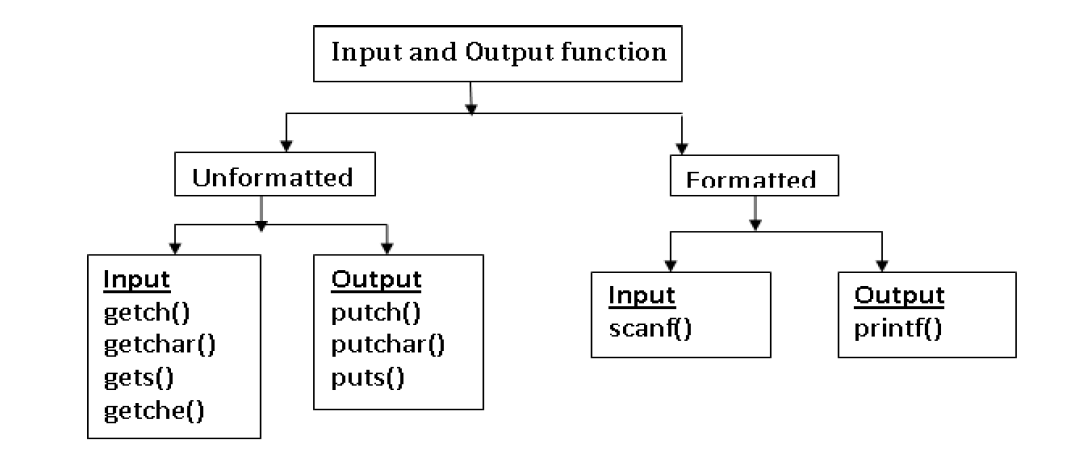
1. Integer constants

2. Floating-point constants

3. Character constants

4. String constants

1. List down the formatted and unformatted I/O functions? **[CO1, K1]**

****

1. Write the syntax for declaring and defining a variable? **[CO1, K1]**

Variable Declaration. A variable declaration is when you specify a type and an identifier but have not yet assigned a value to the variable.

Eg. int a;

Variable Definition. A variable definition is when you assign a value to a variable, typically with the assignment operator =.

Eg. int a=10;

1. Define Data type and its types. **[CO1, K1]**

Data types specify how we enter data into our programs and what type of data

we enter.

These data types have different storage capacities.

Data types are broadly classified as,

1. Primary data types

2. Derived data types

3. User Defined data types

1. Define Operator and Operand. Give example. **[CO1, K2]**

**Operator**

C language supports a rich set of built-in operators.

An operator is a symbol that tells the compiler to perform certain mathematical

or logical manipulations.

Operators are used in program to manipulate data and variables

**Operand**

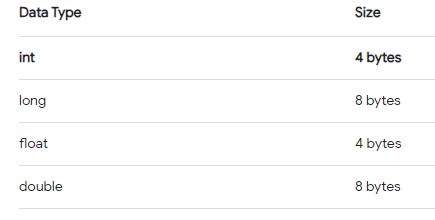
An operand can be a constant, a variable or a function result.

Eg. c=a+b;

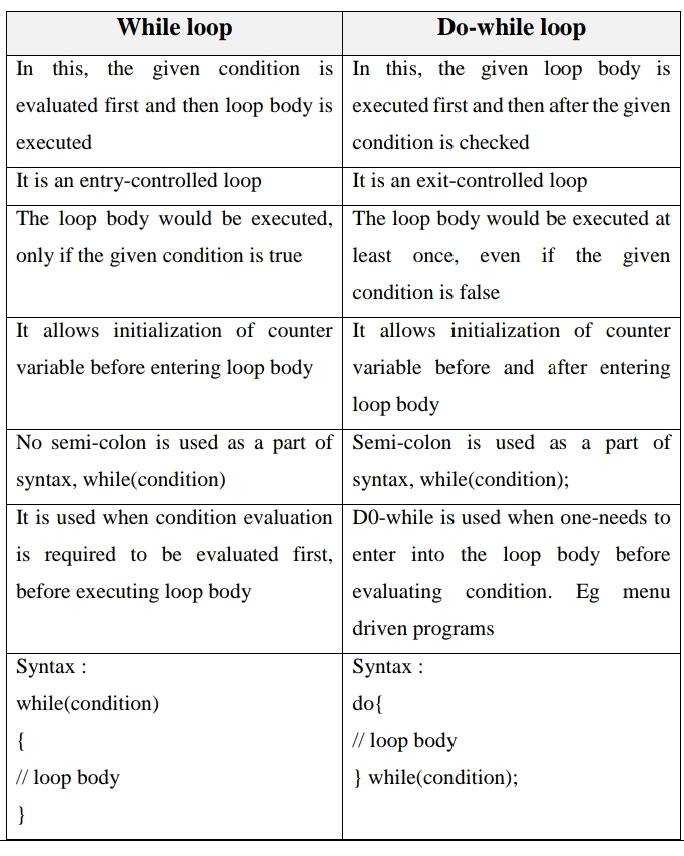
Here: a,b,c are operands

          =,+ are operators

1. How many bytes are occupied by the int, char, float, long int & double? **[CO1, K2]**

****

1. What is the difference between while and do while statement? **[CO1, K2]**

****

1. Differentiate ++a and a++ with an example. **[CO1, K2]**

The difference lies in which value is used in the rest of the expression containing it.         With ++a, a is incremented first, and that is the value you get; with a++ you get the value of a, and a is incremented afterwards.

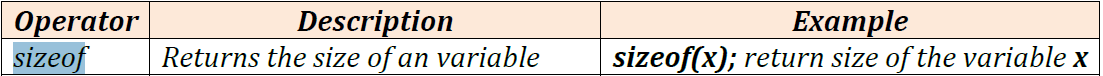
1. What is ternary Operator? **[CO1, K2]**

A conditional operator is also known as ternary operator and used to evaluate conditional expression.

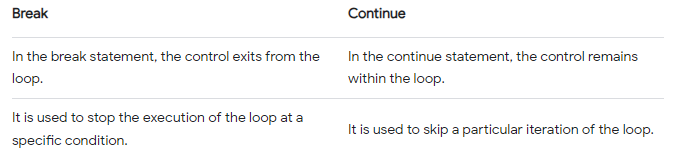
expr1 ? expr2 : expr3

If epr1 Condition is true ? Then value expr2 : Otherwise value expr3

1. What is the use of sizeof operator? **[CO1, K1]**

****

1. Differentiate continue and break in C? **[CO1, K2]**

****

1. What do you mean by expressions. Give examples. **[CO1, K1]**

An expression is a formula in which operands are linked to each other by the use of operators to compute a value. An operand can be a function reference, a variable, an array element or a constant.

Eg 6\*2/ (2+1 \* 2/3 + 6) + 8 \* (8/4)

**PART-B**

1. Discuss in detail about various operators used in C with an example**. [CO1, K2]**
2. Explain briefly the formatted and unformatted I/O functions. **[CO1, K2]**
3. Write about types of Decision-making statements and discuss with example. **[CO1, K2]**
4. Explain a structure of a C program. What are the advantages and applications of C language? **[CO1, K2]**
5. Write a program for the following: **[CO1, K3]**

a) To check whether a number is prime or not.

b) To find the digits of a number.

c) To find the sum of first n integers.

d) To check whether a given number is a Armstrong number or not.

1. Explain the different types of datatypes with example. **[CO1, K2]**
2. Explain about preprocessor directives with examples? **[CO1, K2]**
3. Write a C Program to implement a menu driven calculator using switch statement **[CO1, K3]**

**UNIT II ARRAYS AND STRINGS**

**PART-A**

1. Define an Array and its Types with an Example. **[CO2, K1]**

An array is a collection of similar data elements stored at contiguous memory locations. It can be used to store the collection of primitive data types such as int, char, float, etc., and also derived and user-defined data types such as pointers, structures, etc.

**Syntax of Array Declaration:** data\_type array\_name [size];

**Types of Arrays in C:**

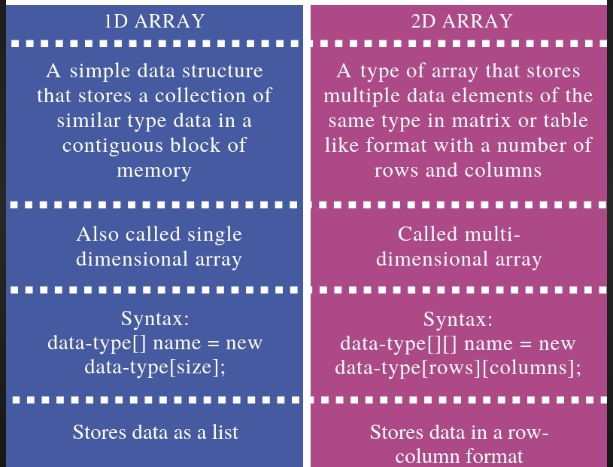
One Dimensional Arrays (1D Array)

Multidimensional Arrays (2D Array)

1. **Write features of an array. [CO2, K1]**

* **Fixed Size:** Arrays have a fixed size that is specified at the time of declaration.
* **Contiguous Memory Allocation:** The elements of an array are stored in adjacent memory locations.
* **Homogeneous Data Type**: All elements of an array must be of the same data type.
* **Random Access:** Elements in an array can be accessed directly using their index.

1. **Difference between one dimensional and two-dimensional array. [CO2, K2]**

****

1. **Why don’t we use the & symbol while reading string through scanf()? [CO2, K2]**

Strings in C are represented as arrays of characters. The name of the string itself points to the base address of the character array. So there is no need to use & in scanf function.

**Example:**

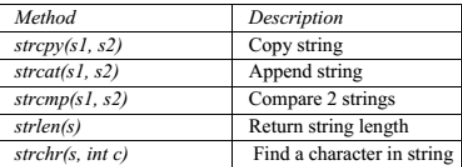
char name[50];

printf("Enter your name: ");

scanf("%s", name); // No & symbol before 'name'

In this code, the name array is already a pointer to the first character of the string, so the function knows that it's dealing with a string and reads characters into the array accordingly.

1. **List out four predefined functions for string manipulation. [CO2, K1]**



1. **Write syntax for string declaration with an example [CO2, K1]**

A String in C programming is a sequence of characters terminated with a null character ‘\0’. The C String is stored as an array of characters.

**Syntax: char string\_name[size]; Or**

**char string\_name[size] = "string";**

char c[] = "abcd";

char c[50] = "abcd";

char c[] = {'a', 'b', 'c', 'd', '\0'};

char c[5] = {'a', 'b', 'c', 'd', '\0'};

1. **What is sorting? and why it is necessary? [CO2, K2]**

Sorting in c is the processing of arranging the data in ascending and descending order. There are several types of sorting in data structures, namely – bubble sort, insertion sort, selection sort, bucket sort, heap sort, quick sort, radix sort, etc.

By sorting data, it is easier to search through it quickly and easily. The simplest example of sorting is a dictionary.

1. **What is linear search? List advantages and disadvantages. [CO2, K2]**

Linear search, also known as a sequential search, is a straightforward searching algorithm that is used to find a specific element within a list or array.

It works by examining each element in the list one by one until a match is found.

**Algorithm:**

Start from the beginning of the list.

Compare the target element with each element in the list, in order.

If a match is found, return the index (position) of the element. If the entire list is searched and no match is found, return a sentinel value (e.g., -1) to indicate that the element is not in the list.

**Advantages of Linear Search:**

* Simplicity
* Applicability
* No pre-processing

1. **What is binary search? List advantages and disadvantages [CO2, K2]**

Binary search is a search algorithm that is based on the principle of divide-and-conquer strategy and is used to efficiently locate a target element by repeatedly dividing the search space in half until the desired element is found.

**Algorithm:**

Start with the entire sorted array.

Compare the target element with the middle element of the array.

If the target element matches the middle element, the search is successful.

If the target element is less than the middle element, continue the search in the left half of the array.

If the target element is greater than the middle element, continue the search in the right half of the array.

**Advantages:**

* Efficiency
* Applicability
* Optimization

1. **How can you assign one array to another array? [CO2, K2]**

#include <stdio.h>

  int main()

{

      int a[5] = { 3, 6, 9, 2, 5 }, n = 5;

    int b[n], i;

      // copying elements from one array to another

    for (i = 0; i < n; i++)

 {

         b[i] = a[i];

    }

     printf("\nThe copied array is :");

    for (i = 0; i < n; i++)

{

         printf("%d ", b[i]);

     }

    return 0;

}

Output

The copied array is :3 6 9 2 5

1. **Write the features of string [CO2, K1]**

* **Array of Characters:** Strings are essentially arrays of characters.
* **Null-Termination:** Strings in C are terminated with ‘\0’.
* **Library Functions:** C provides a set of standard library functions
* Individual characters in a string can be accessed using array notation.
* Strings can be initialized at the time of declaration.

1. **What is the difference between gets () and scanf () while reading a string. [CO2, K2]**

**gets():**

* It is a standard library function declared in <stdio.h>.
* Reads a line of text from the standard input (usually the keyboard) until it encounters a newline character ('\n'). It includes spaces in the input.

**scanf():**

* It is a more general-purpose function for reading input and is also declared in <stdio.h>.
* Reads input based on the format specifier. It stops reading when it encounters whitespace (space, tab, newline) and doesn't include spaces in the input string.

1. **Write a program to find the length of the string. [CO2, K2]**

#include <stdio.h>

#include <string.h>

int main()

{

char Str[1000];

int i;

printf("Enter the String: ");

scanf("%s", Str);

printf("Length of Str is %ld", strlen(Str));

return 0;

}

1. **Advantages and disadvantages of an array. [CO2, K1]**

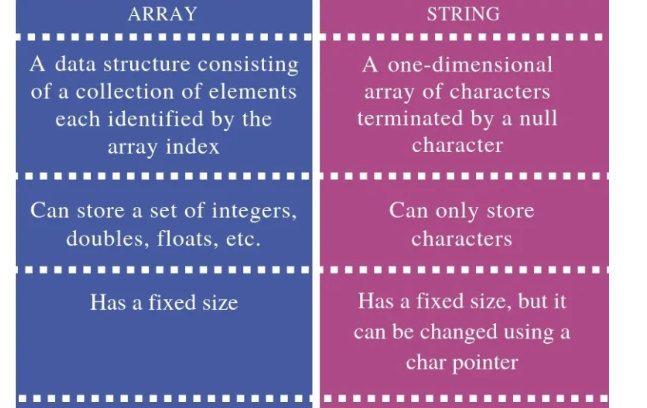
**Advantages of Arrays:**

* Efficient Access of elements
* Sequential Memory Allocation
* Easy to Implement and Use
* Fixed Size
* Memory Efficiency

**Disadvantages of Array:**

* Inefficient Insertion/Deletion
* Wasteful Memory Allocation
* Lack of Dynamic Sizing
* Fixed Size

1. **What is the difference between a string and an array? [CO2, K2]**

****

**PART-B**

1. Explain in detail about the predefined string handling functions with syntax and example. **[CO2, K2]**
2. Writes notes on arrays and its types with example. **[CO2, K2]**
3. Explain about linear search with an example. **[CO2, K2]**
4. Explain about binary search with an example. **[CO2, K2]**
5. What is meant by sorting. Write a C program to implement selection sort. **[CO2, K2]**
6. Implement matrix addition & multiplication using arrays **[CO2, K2]**
7. Write a C program **[CO2, K2]**
8. To print an array in a reverse order.
9. To check whether the given string is palindrome or not.
10. To count no of occurrences of a particular character in a string.
11. How will you find the max, min, second largest, second smallest and middle element of an array in a single program. **[CO2, K3]**

**UNIT-III FUNCTIONS AND POINTERS**

**PART A**

**1.What is the need for functions?[CO3-K2]**

**·** To reduce the complexity of large programs

·        To increase the readability

·        To achieve reusability

·        To avoid redundancy of code

·        To save Memory

**2.Compare actual parameter & formal parameter.[CO3-K2]**

Actual Parameter: Specified in the function call statement. Used to supply the input values to the function either by copy or reference .

Formal Parameter: Specified in the function definition statement. It takes either copy or address of the actual arguments.

**3.Differentiate call by value and call by reference.[CO3-K2]**

Call by value: The values of the variables are passed by the calling function to the called function.

Call by reference: The addresses of the variables are passed by the calling function to the called function.

**4. What is meant by Recursive function?[CO3-K2]**

 If a function calls itself again and again, then that function is called Recursive function.

 Example:

 void recursion()

 {

recursion(); /\* function calls itself \*/

}

 int main()

 {

recursion();

}

**5. Write the advantages of pointer.[CO3-K1]**

Pointer is used in the following cases.

· It is used to access array elements.

· It is used for dynamic memory allocation.

· It is used in call by reference.

· It is used in data structure like trees, graph, etc.

**6. How to declare pointer variable?[CO3-K2]**

A pointer declaration consists of a base type, an \*, and the variable name.     The general form for declaring a pointer variable is

                              data\_type \*var\_name;

  For example

                         int  \*p;

**7.List out any 4 math functions.[CO3-K1]**

pow(x,y) : used to find power of value of xy .

 log10(x) : used to find natural logarithmic value of x

 sqrt(x) : used to find square root vale of

x sin(x) : returns sin value of x

**8.write the syntax for function declaration.[CO3-K2]**

A function declaration usually contains the function name , return type , and the parameter types.

 The following is the syntax for defining a function in C

return\_type function\_name(parameter\_list);

 return\_type is the data type of the value that the function returns.

**9. What is meant by library function?[CO3-K1]**

Library functions are built-in functions that are grouped together and placed in a common location called library. Each function here performs a specific operation. We can use this library functions to get the pre-defined output. All C standard library functions are declared by using many header files.

**10. What are the various parts of functions?[CO3-K2]**

**·**  Input parameters.

·         Output parameters.

·         Return value.

·         Math section.

**11. Write the difference between pre-defined (library) function and user defined**

**function.[CO3-K2]**

|  |  |
| --- | --- |
| (Pre-defined) function | User defined function |
| Contains Pre-defined set of functions | The user defined the functions |
| User cannot understand the internal working | User can understand internal working |
| Source code is not visible | Source code is visible |
| User cannot modify the function | User can modify the function |

**12. Write the Syntax of function call.[CO3-K2]**

Syntax of function call

functionName(argument1, argument2, ...);

function call is made using addNumbers(n1,n2); statement inside the main().

**13. What are the uses of Pointers?[CO3-K2]**

**·**       Pointers are used to return more than one value to the function

·        Pointers are more efficient in handling the data in arrays

·        Pointers reduce the length and complexity of the program

·        They increase the execution speed

·        The pointers save data storage space in memory

**14. Write definition of function. Indicate types of functions available in C.**

**A function is a self-contained block or a sub-program of one or more statements that performs a special task when called.[CO3-K2]**

**·**     Without arguments or return values. Eg. abc()

·        With arguments but without return values. Eg. abc (int x)

·        With arguments and return values. Eg. int abc(int x)

·        Without argument but with return values. Eg. int abc().

**15.Write the advantages and disadvantages of recursion.[CO3-K2]**

Recursion makes program elegant and cleaner. All algorithms can be defined recursively which makes it easier to visualize and prove. If the speed of the program is vital then, you should avoid using recursion. Recursions use more memory and are generally slow. Instead, you can use loop.Check out these examples to learn more:

·        Find the Sum of Natural Numbers using Recursion

·        Find Factorial of a Number Using Recursion

·        Find G.C.D Using Recursion

**PART B**

1.Write a C program to find the factorial of a number using recursion. **[CO3-K3]**

2.What is function? Explain different classification of user defined functions   based on parameter passing and return type with examples. **[CO3-K3]**

3.Differentiate between call by value and call by reference with examples. **[CO3-K3]**

4. Explain function call, function definition and function prototype with examples **[CO3-K3]**

5. What are actual parameters and formal parameters? Illustrate with example

**[CO3-K3]**

6. Discuss about the classification of functions depending upon their inputs and

output (parameters). **[CO3-K3]**

7. Discuss about passing arrays to function? **[CO3-K3]**

8. Explain in detail about recursive function with binary search **[CO3-K3]**

**UNIT IV - STRUCTURES AND UNIONS**

**PART A**

1. Compare arrays and structures. **[CO4, K2]**

**Comparison of arrays and structures is as follows.**

|  |  |
| --- | --- |
| **Arrays** | **Structures** |
| **An array is a collection of data items of same data type. Arrays can only be declared.** | **A structure is a collection of data items of different data types. Structures can be declared and defined.** |
| **There is no keyword for arrays.** | **The keyword for structures is struct.** |
| **An array cannot have bit fields.** | **A structure may contain bit fields.** |
| **An array name represents the address of the starting element.** | **A structure name is known as tag. It is a Shorthand notation of the declaration.** |

1. Compare structures and unions. **[CO4, K2]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Structure** | **Union** | |  |
| **Every member has its own memory.** | | | **All members use the same memory.** | |
| **The keyword used is struct.** | | | **The keyword used is union.** | |
| **All members occupy separate memory location, hence different interpretations of the same memory**  **location are not possible. Consumes more space compared to union.** | | | **Different interpretations for the same memory location are possible. Conservation of memory is**  **possible** | |
|  |  |  |  |  |

1. Define Structure in C. **[CO4, K1]**

* **C Structure is a collection of different data types which are grouped together and each element in a C structure is called member.**
* **If you want to access structure members in C, structure variable should be declared.**
* **Many structure variables can be declared for same structure and memory will be allocated for each separately.**
* **It is a best practice to initialize a structure to null while declaring, if we don‘t assign any values to structure members.**

1. How to Declare a member in Structure? **[CO4, K2]**

**A struct in C programming language is a structured (record) type[1] that aggregates a fixed set of labeled objects, possibly of different types, into a single object. The syntax for a struct declaration in C is:**

**struct tag\_name**

**{**

**Type   attribute;**

**type attribute2;**

***/\* ... \*/***

**};**

1. What is meant by Union in C? **[CO4, K1]**

**A union is a special data type available in C that enables you to store different data types in the same memory location. You can define a union with many members, but only one member can contain a value at any given time. Unions provide an efficient way of using the same memory location for multi-purpose.**

1. How to initialize the member of union in C. **[CO4, K2]**

**To define a union, you must use the union statement in very similar was as you did while defining structure. The union statement defines a new data type, with more than one member for your program. The format of the union statement is as follows:**

**union [union tag]**

**{**

**Data members;**

**---**

**---**

**} [one or more union variables];**

1. What are storage classes? **[CO4, K1]**

**A storage class defines the scope (visibility) and life time of variables and/or functions within a C Program.**

1. What are the storage classes available in C? **[CO4, K1]**

**There are following storage classes which can be used in a C Program**

**1.  auto**

**2.  register**

**3.        static**

**4.        extern**

1. What is register storage in storage class? **[CO4, K1]**

**Register is used to define local variables that should be stored in a register instead of**

**RAM. This means that the variable has a maximum size equal to the register size (usually one word) and cant have the unary '&' operator applied to it (as it does not have a memory location).**

**{**

**register int Miles;**

**}**

1. What is static storage class? **[CO4, K1]**

**Static is the default storage class for global variables. The two variables below (count and road) both have a static storage class.**

**static int Count; int Road;**

**{**

**printf("%d\n", Road);**

**}**

1. Define Auto storage class in C. **[CO4, K1]**

**auto is the default storage class for all local variables.**

**{**

**int Count; auto int Month;**

**}**

**The example above defines two variables with the same storage class. auto can only be used within functions, i.e. local variables.**

1. Define singly linked list. **[CO4, K1]**

**A singly linked list is a special type of linked list in which each node has only one link that points to the next node in the linked list.**

1. What is the use of ‘typedef’’? **[CO4, K1]**

**It is used to create a new data using the existing type.**

**Syntax: typedef data type name;**

**Example:**

**typedef int hours: hours hrs;/\* Now, hours can be used as new datatype \*/**

1. Write the syntax for pointers to structure. **[CO4, K1]**

**Struct S**

**{**

**char datatype1; int datatype2; float datatype3;**

**};**

**Struct S \*sptr; //sptr ia pointer to structure S**

1. What are the various dynamic memory allocation functions? **[CO4, K1]**

* **malloc() - Used to allocate blocks of memory in required size of bytes.**

**•     free () - Used to release previously allocated memory space.**

**•     calloc() - Used to allocate memory space for an array of elements.**

**•     realloac() - Used to modify the size of the previously allocated memory space.**

**PART B**

1. Create a structure with data members of various types and declare two structure variables. Write a program to read data into these and print the same. Justify the need for structured data type. **[C04, K3]**
2. Develop a C program to store the employee information using structure and search a particular employee using Employee number. **[CO4, K3]**
3. Explain about array of structures and pointers in structures with example program. **[CO4, K2]**
4. Discuss and develop a C program about dynamic memory allocation with suitable example. **[CO4, K3]**
5. Explain about singly linked list with suitable example C program and perform the operations for insertion in a list. **[CO4, K3]**
6. Develop a C program to store the student information using union. **[CO4, K3]**
7. Explain about singly linked list with suitable example C program and perform the operations for deletion operation in a list**. [CO4, K3]**
8. Develop a C program to store the student information with data members student id, name, marks for three subjects, total and average using structure. **[CO4, K3]**

**UNIT V - FILE PROCESSING**

**PART A**

1. What is File handing in C? **[CO5, K1]**
2. Why files are needed in C programming language? **[CO5, K2]**
3. Mention the features of File Handling in C? **[CO5, K2]**
4. List the Operations that can be carried out on files in C. **[CO5, K1]**
5. What are the types of files in C? **[CO5, K1]**
6. Write a note on text file and binary file. **[CO5, K1]**
7. What is File Pointer in C? **[CO5, K1]**
8. How to Open a File in C? **[CO5, K2]**
9. What is read and write in file operation**? [CO5, K1]**
10. Difference between Append Mode and Write Mode. **[CO5, K2]**
11. How Sequential access file differ from Random access file. **[CO5, K2]**
12. What do you understand by Sequential access file? **[CO5, K1]**
13. What are the advantages of sequential access files? **[CO5, K1]**
14. What types of applications are best suited for random access files? **[CO5, K1]**
15. What are Command line arguments? **[CO5, K1]**

**PART - B**

1. Explain in detail on File handling in C with examples. **[CO5, K3]**
2. Explain in detail various operations that can be done on file with suitable examples. **[CO5, K3]**
3. Discuss about the types of file processing in C. **[CO5, K3]**
4. Write a C program to find the average of numbers stored in sequential file access. **[CO5, K3]**
5. Describe on Random access in files along with the functions used for the same in C. Give suitable example. **[CO5, K3]**
6. Elaborate on Command line arguments with an example program. **[CO5, K3]**
7. Write a detailed note on the comparison of Sequential access file and Random-access file. **[CO5, K3]**

Elaborate on Command line arguments with an example program. **[CO5, K3]**