CONTENT 2

### Basic Structure of C Program

 We will be discussing some of the structural parts of our code, such as:

* Pre-processor commands
* Functions
* Variables
* Statements
* Expressions
* Comments

#### Here I am providing a Simple code, so you can understand certain parts of program. Code as described/written

#include <stdio.h>

int main()

{

int a, b;

printf("Enter number a\n");

scanf("%d", &a);

printf("Enter number b\n");

scanf("%d", &b);

printf("The sum is %d\n", a+b);

return 0;

}

Now we will divide our written program into a few lines of code and one by one we will go over the meaning of each and every keyword in the specific line.

So, let’s begin with **Pre-processor commands**.

#include <stdio.h>

This is the first line of our code. Any line starting with # represents a preprocessing command. It tells our program that before its execution, it must include the stdio.h named file in it because we are using some of the commands or codes from this file

**For example**, if our programs needs mathematical operations of high level then we must include:

#include <math.h>

It helps us to use the code from math.h file for the calculations in our programs.

int main()

this is the 2nd line. main() is **function** here and we will see the detail about the function in upcoming or later Document. Here int is the **return type** of function and the return type is according to the functions activity i.e. if it is giving an integer variable as a result then return type should be int.

int a,b;

here we are **initializing** two variables as integers. Initializing with integer means that we can store integer values in it. If we would have initialized them with char then we could have stored character values in it such as a, b, c, d, etc.

printf("Enter number a\n");

This is simply a **print statement**. Whatever we write in the brackets will be directly printed onto the screen. /n is the **new line character** here.

scanf("%d", &a);

 scanf is used to take **inputs from the user**. Here &a gives the address of variable “a” to store the user's given value. %d notifies that the value should be of integer type.

printf("The sum is %d\n", a+b);

Here a+b is simply a **mathematical addition** and  print statement is printing the result onto the screen.

return 0;

we need a **return value** for a function. The function we created was of int type so it is returning 0. Return 0 means that the function is working perfectly.

Note: Return statement and function type should be same.

//This is a comment

We write **comments** by using the double slash sign (//). Comments are to notify other programmers the working of the code at specific intervals or we write them for our-self. They do not have any effect on the written program.

**Now we are going to write a command in our power terminal window, down below:**

gcc-Wall-save-temps file\_name.c -o new\_file

**Note:** Here the **file\_name** is the name of the file we created to write code and **new\_file** is the name we want our executable file to have.

Now I will tell you the benefits of this commandv in VScode terminal. By the help of this command five files will be created namely:

* file\_name.i
* file\_name.s
* file\_name.o
* file\_name.c
* new\_file.exe

**file\_name.i** will create a **preprocessing file** where comments are removed, macros are expended and all the code from # files are copied into the file\_name.i file with our respective code at the end.

**file\_name.s** will have our code converted in **assembly** level code

**file\_name.o** will have all the code **file\_name.c** will contain our **executable C language code** in  **machine level language** in binary form.

**new\_file.exe** is the linker that**links** all the file\_name.o sort of files at on place.

**“And this some of the advantages or features of an IDE that we can also get deep knowledge and working about the codes that had been executed.”**

Code as described/written

#include <stdio.h>

int main()

{

printf("Hello World\n");

return 0;

}