(MA) (10)

* **Command line Argument:-** It is a parameter supplied to the program when it is invoked. It is mostly used when you need to control your program from outside. Command line arguments are passed to the main() method.

Syntax:

```
int main(int argc, char *argv[])
```

→ Here `argc` counts the number of arguments on the command line and `argv[]` is a pointer array which holds pointers of type `char` which points to the arguments passed to the program.

→ **Program for Command line argument:**

```
#include <stdio.h>
#include <conio.h>
int main(int argc, char *argv[])
{
    int i;
    if(argc >=2)
    {
  Printf("The argument supplied are:");
    for(i=0; i<argc; i++)
    {
     Printf("%s", argv[]);
    } }
  else {
  Printf("argument list is empty");
  }
  return 0;
}
```

→ Remember that argv[0] holds the name of the program and argv[1] points to the first Command line argument and argv[n] gives the last argument. If no argument is supplied, argc will be 1.

→ argc (ARgument Count):- is int and stores no. of Command-line arguments passed by the user including the name of the program. so if we pass a value to a program, value of argc would be 2 (one for argument and one for program name).

↳ The value of argc should be non-negative.

↳ argv (ARgument Vector) is array of characters pointer listing all the arguments.

↳ if argc is greater than zero, the array elements from argv[0] to argv[argv-1] will Contain pointers to strings.

↳ Argv[0] is the name of the program, After that till argv[argc-1] every element is Command-line arguments.

* Properties of Command line Arguments:

1. They are passed to main() function.
2. They are parameters/arguments supplied to the program when it is invoked.
3. They are used to Control program from outside instead of hard coding those values inside the code.
4. argv[argc] is a NULL Pointer.
5. argv[0] holds the name of the program.
6. argv[1] points to the first Command line argument and argv[n] points last arguments.