LAB 1
PROGRAMMING FUNDAMENTALS WITH C

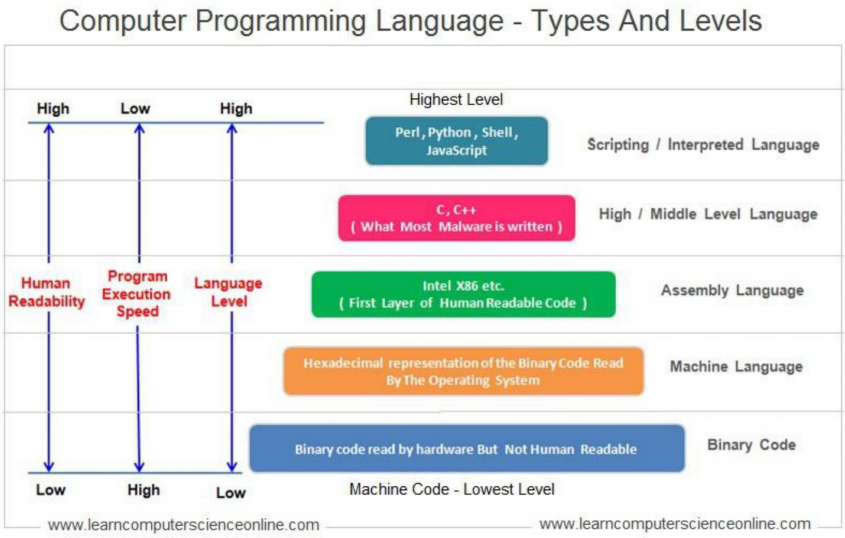
BACKGROUND:
What is Operating System?

An operating system (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. The operating system is a vital component of the system software in a computer system.

Main Task of Operating System

- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users

All the major operating systems are written in C. The language of Microsoft Windows at API level is C. Mac OS X, which is based on Unix, is written in C.



INTRODUCTION TO C:

C programming is a general-purpose, procedural, imperative computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system. C is the most widely used computer language. It keeps fluctuating at number one scale of popularity along with Java programming language, which is also equally popular and most widely used among modern software programmers.

- Easy to learn
- Structured language
- It produces efficient programs
- It can handle low-level activities
- It can be compiled on a variety of computer platforms

Facts about C

- C was invented to write an operating system called UNIX.
- C is a successor of B language, which was introduced around the early 1970s.
- The American National Standard Institute (ANSI) formalized this language in 1988.
- The UNIX OS was totally written in C.
- Today C is the most widely used and popular System Programming Language.
- Most of the state-of-the-art software has been implemented using C.
- Today's most popular Linux OS and RDBMS MySQL have been written in C.

Why use C

C was initially used for system development work, particularly the programs that make-up the operating system. C was adopted as a system development language because it produces code that runs nearly as fast as the code written in assembly language. Some examples of the use of C are -

- Operating Systems
- Language Compilers
- Assemblers
- Text Editors
- Print Spoolers
- Network Drivers
- Modern Programs
- Databases
- Language Interpreters
- Utilities

C Programs

A C program can vary from 3 lines to millions of lines and it should be written into one or more text files with extension ".c"; for example, hello.c

If you want to set up your environment for C programming language, you need the following two software tools available on your computer, (a) Text Editor and (b) The C Compiler.

Example: 1.1

```
#include<stdio.h>
Int main(){
        /* my first program in C */
        printf("Hello, World! \n");
        return 0;
}
```

Input and Output In C

Output:
printf("%d",variable_name);

Input:
scanf("%d",&variable_name);

Format Specifiers in C

It is a way to tell the compiler what type of data is in a variable during taking input using scanf() or printing using printf().

Format Specifier	Description
%d	Integer Format Specifier
%f	Float Format Specifier
%c	Character Format Specifier
%s	String Format Specifier
%u	Unsigned Integer Format Specifier
%ld	Long Int Format Specifier

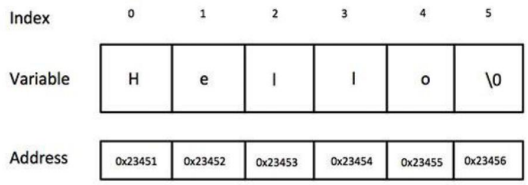
Example 1.2: Write a program that takes two number and add them

```
#include<stdio.h>
int main(){
int a,b,c;
printf("Enter two numbers: \n");
scanf("%d %d",&a,&b);
c = a+ b;
printf("The addition of two number is: %d", c);
return0;
```

Strings in C

Strings are defined as an array of characters. The difference between a character array and a string is the string is terminated with a special character '\0'.

Following is the memory presentation of the above defined string in C/C++ –



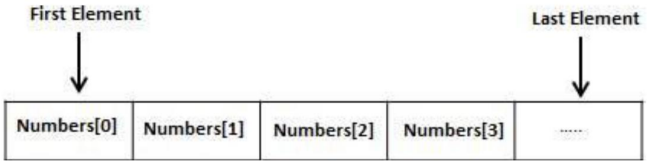
Declaration Of Strings: char str_name[size];
Reading String: scanf("%s",str);
Print String: printf("%s",str);

Example 1.3: Write a program that takes name as input and prints it.

```
#include<stdio.h>
int main() {
   char name[20];
   printf("Enter name: ");
   scanf("%s", name);
   printf("Your name is %s.", name);
   return 0;     }
```

Arrays in C

Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type. All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.



For example, if you want to store 100 integers, you can create an array for it.

dataType arrayName[arraySize];

Example:

```
int books[3] = {19, 10, 8};
OR
int books[] = {19, 10, 8};
OR
int books[3];
```

Example 1.4: Program to find the average of 5 numbers using arrays.

```
#include<stdio.h>
int main() {
int arr[5],num,i;          //array declaration
double avg=0,sum=0; //variable declaration
printf("Enter the numbers: \n");
for(i=1; i<=num; i++){
scanf("%d", &arr[i]);
}
for(i=1; i<=num; i++){
sum=sum+arr[i];          //loop for calculating sum
}
avg=sum/num;                    //calculate average
printf("Average of entered numbers are: %f",avg);
return 0;          }
```

Functions in C

A function is a group of statements that together perform a task. Every C program has at least one function, which is main(), and all the most trivial programs can define additional functions.

Defining a Function

Function Declaration

```
return-type function-name(parameter list);
```

Function Definition

```
return-type function-name( parameter list ) {
  body of the function
}
```

Calling A Function

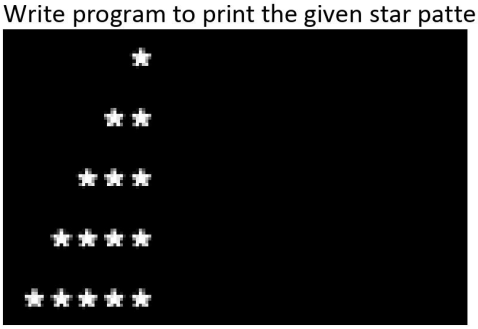
```
function-name (parameter list);
```

Example 1.5: Create a function that takes numbers in array as perimeter and returns the greatest number and smallest.

```
#include<stdio.h>
#include <conio.h>
 void max_min(int a[],int n){
        int min,max,i;
        min=max=a[0];
        for(i=1; i<n; i++){
                if(min>a[i])
                        min=a[i];
                if(max<a[i])
                        max=a[i];      }
        printf("minimum of array is : %d",min);
        printf("\nmaximum of array is : %d",max);
 }
int main(){
   int a[100],i,n,sum;
   printf("Enter size of the array : ");
   scanf("%d", &n);
   printf("Enter elements in array : ");
   for(i=0; i<n; i++){
      scanf("%d",&a[i]);          }
   max_min (a,n);
}
```

Lab-1 Exercises

1. Write a program that multiply and divide two numbers.
2. Write program to print the given star pattern



3. Write C Program to Multiply Two Matrices Using Multi-dimensional Arrays