LECTURE 1

c programming
(variables, keywords, data types, loop, one dimensional array)

by
Md Sahid Alam
Gour Mavidyalaya
22.12.2024

Structure of C Language program

Preprocessor directive (Header file)
Global variable declaration (User defined function)
main function( )
{

Local variables;
Statements;

}

or(with more details)

Comment line
Preprocessor directive (Header file)
Global variable declaration
main function( )
{

Local variables;
Statements;

}
User defined function

Comment line
It indicates the purpose of the program. It is represented as
/*.................................*/
or
//...............................
Comment line is used for increasing the readability of the program. It is useful in explaining the program and generally used for documentation. It is enclosed within the decimeters. Comment line can be single or multiple line but should not be nested. It can be anywhere in the program except inside string constant & character constant.

Preprocessor Directive:

#include<stdio.h> tells the compiler to include information about the standard input/output library. It is also used in symbolic constant such as #define PI 3.14(value). The stdio.h (standard input output header file) contains definition &declaration of system defined function such as printf( ), scanf( ), pow( ) etc. Generally printf() function used to display and scanf() function used to read value.

Common Standard Library Header Files in C:
1. <stdio.h> — Standard Input/Output Library Functions for input/output operations like printf(), scanf(), getchar(), and putchar().

2. <stdlib.h> — Standard Library Functions for memory allocation (malloc(), free()), process control (exit()), and conversions (atoi(), atof()).

3. <math.h> — Mathematical Functions Provides mathematical functions like sqrt(), pow(), sin(), cos(), etc.

4. <string.h> — String Manipulation Functions for manipulating strings like strlen(), strcpy(), strcmp(), and strcat().

5. <ctype.h> — Character Handling Functions for checking character types and converting characters, such as isalnum(), isdigit(), toupper(), etc.

6. <time.h> — Time and Date Functions for time and date operations, such as time(), clock(), localtime(), and strftime().

7. <limits.h> — Limits of Data Types Defines constants for the limits of integral types (e.g., INT_MAX, LONG_MIN, CHAR_BIT).

8. <assert.h> — Assertion Library

Global Declaration:
This is the section where variable are declared globally so that it can be access by all the functions used in the program. And it is generally declared outside the function.

Main function
It is the user defined function and every function has one main() function from where actually program is started and it is encloses within the pair of curly braces. The main( ) function can be anywhere in the program but in general practice it is placed in the first position.

Syntax :

```
 int main()            main()            void main()
 {                     {                   {
   ...                  ...                 ...
   ...                  ...                 ...
return 0;               ...                  ...
     }                   }                   }
```

EX: Writer a C program to display "Hello! world"

```
#include <stdio.h>
int main()
{
printf ("Hello! world\n");
return0;
}
```

Output: Hello! world
EX: Write a C program to add two numbers.

```
#include <stdio.h>
int main()
{
int a, b, sum;
printf("Enter two numbers: ");
scanf("%d%d",&a,&b);
sum = a + b;
printf ("The sum of %d and %d is %d\n", a, b, sum);
return 0;
}
```

Output:

```
Enter two numbers: 12 45
The sum of 12 and 45 is 57
```

Keyword

There are certain words reserved for doing specific task, these words are known as **reserved word** or **keywords.** These words are predefined and always written in lower case or small letter. These keywords cann't be used as a variable name as it assigned with fixed meaning. Some examples are

Keyword	**Description**
`auto`	Used to declare automatic variables.
`break`	Exits from a loop or switch statement.
`case`	Defines a branch in a switch statement.
`char`	Defines a character data type.
`const`	Defines a constant value.
`continue`	Skips the current iteration of a loop and continues with the next iteration.
`default`	Specifies the default case in a switch statement.
`do`	Starts a do-while loop.
`double`	Defines a double-precision floating-point number.
`else`	Specifies an alternative branch in an if-else statement.
`enum`	Defines an enumerated type.
`extern`	Declares an external variable or function.
`float`	Defines a floating-point number.
`for`	Starts a for loop.
`goto`	Transfers control to a specified label in the code.
`if`	Starts an if statement.
`int`	Defines an integer data type.
`long`	Defines a long integer data type.
`register`	Suggests the compiler to store the variable in a register.
`return`	Exits from a function and optionally returns a value.
`short`	Defines a short integer data type.

Keyword	**Description**
`signed`	Specifies a signed integer data type.
`sizeof`	Returns the size, in bytes, of a data type or object.
`static`	Defines a static variable or function.
`struct`	Defines a structure.
`switch`	Defines a switch statement for branching.
`typedef`	Defines a new type name.
`union`	Defines a union data type.
`unsigned`	Specifies an unsigned integer data type.
`void`	Represents no type or an empty type.
`volatile`	Indicates that a variable's value may change at any time.
`while`	Starts a while loop.

Variables

Variable is a data name which is used to store some data value or symbolic names for storing program
computations and results. The value of the variable can be change during the execution. The rule for naming the variables is same as the naming identifier. Before used in the program it must be declared. Declaration of variables specify its name, data types and range of the value that variables can store depends upon its data types.

Syntax:
int a;
char c;
float f;
Variable initialization
When we assign any initial value to variable during the declaration, is called initialization of variables. When variable is declared but contain undefined value then it is called garbage value. The variable is initialized with the assignment operator such as
Data type variable name=constant;
Example: int a=20;
Or int a;
a=20;

Data types

Data types refer to an extensive system used for declaring variables or functions of different types before its use. The type of a variable determines how much space it occupies in storage and how the bit pattern stored is interpreted. The value of a variable can be changed any time. C has the following 4 types of data types

basic built-in data types: int, float, double, char

Enumeration data type: enum

Derived data type: pointer, array, structure, union

Void data type: void

Basic data type	**Example**	**Format specifiers**

`int`	int a;
	int a,b=1,c;
	int number;
		%d

`float`	float a;
	float a,b=3.0,c;
	float number;
		%f

`double`	double a;
	double a,b,c;
	double value_1;
		%lf

`char`	char a;
	char a='A',b,c='*'c
	char letter='M';
		%d

Increment and Decrement

The Unary operator ++, --, is used as increment and decrement which acts upon single operand. Increment operator increases the value of variable by one.
Similarly decrement operator decrease the value of the variable by one.And these operator can only used with the variable, but cann't use with expression and constant as ++6 or ++(x+y+z).
It again categories into prefix post fix . In the prefix the value of the variable is incremented 1st, then the new value is used, where as inpostfix the operator is written after the operand(such as a++,a--).

Example:
let y=5;
z= ++y;

```
y= y+1;
z= y;
```

Similarly in the postfix increment and decrement operator is used in the operation . And then increment and decrement is perform.

Example:

```
let x= 5;
y= x++;
y=x;
x= x+1;
```

Loops in C

Loop:-it is a block of statement that performs set of instructions. In loops Repeating particular portion of the program either a specified number of time or until a particular no of condition is being satisfied.

There are three types of loops in c

- While loop
- do while loop
- for loop

While loop :

Syntax:

```
    while(condition)
    {
    Statement 1;
    Statement 2;
    }
```

Example: Write a C program to calculate sum from 1 to 10 using while loop.

```
include <stdio.h>
int main()
{
int sum = 0;
int i = 1;
while (i <= 10)
{
sum += i;
i++;
}
printf("The sum of numbers from 1 to 10 is: return 0;
}
```

Output:

```
The sum of numbers from 1 to 10 is: 55
```

do while loop :

This (do while loop) statement is also used for looping. The body of this loop may contain single statement or block of statement.

Syntax:

```
    do
    {
    Statement;
    }
while(condition);
```

Example: Write a C program to calculate sum from 1 to 10 using do while loop.

```
include <stdio.h>
int main()
{
int sum = 0;
int i = 1;
do
{
sum += i;
i++;
} while (i <= 10);
printf("The sum of numbers from 1 to 10 is: return 0;
}
```

Output:

```
The sum of numbers from 1 to 10 is: 55
```

for loop :

Syntax:

```
    for(exp1;exp2;exp3)
    {
    Statement;
    }
```

Or

```
    for(initialized counter; test counter; update counter)
    {
    Statement;
    }
```

Example: Write a C program to calculate sum from 1 to 10 using for loop.

```
include <stdio.h>
int main()
{
int sum = 0;
for (int i = 1; i <= 10; i++)
{
sum += i;
}
printf("The sum of numbers from 1 to 10 is: return 0;
}
```

Output:

```
The sum of numbers from 1 to 10 is: 55
```

Array

Array is the collection of similar data types or collection of similar entity stored in contiguous memory location. Array of character is a string. Each data item of an array is called an element. And each element is unique and located in separated memory location. Each of elements of an array share a variable but each element having differentindex no. known as subscript. An array can be a single dimensional or multi-dimensional and number of subscripts determines its dimension. And number of subscript is always starts with zero. One dimensional array is known as vector and two dimensional arrays are known as matrix.

Adantages: array variable can store more than one value at a time where other variable can store one value at a time.

Example:

int a[30]; int mark[100];

Syntax:

```
Data type array name [size];
int a[35];
int mark[100];
int a[5]={10,20,30,100,5}
```

Example: Write a C program to add ten numbers using array.

```
#include<stdio.h>
int main()
{
int i ;
int arr [15];
int sum=o;
for (i=0; i<=9; i++)
{
printf (‘Enter a value for arr[i] = \n”, i+1);
scanf (“%d”, &arr[i]);
}
for (i=0; i<=9; i++)
{
sum = sum + a[i];
}
printf ("The sum of ten numbers is %d”, sum);
return 0;
}
```

OUTPUT:

```
Enter a value for arr[0] =5
Enter a value for arr[1] =10
Enter a value for arr[2] =15
Enter a value for arr[3] =20
Enter a value for arr[4] =25
Enter a value for arr[5] =30
Enter a value for arr[6] =35
Enter a value for arr[7] =40
Enter a value for arr[8] =45
Enter a value for arr[9] =50
The sum of ten numbers is 275
```

Example: Write a C program to find sum of three numbers.

```
include <stdio.h>
int main()
{
int a,b,c,sum;
printf("Enter three numbers: ");
scanf("%d %d %d", &a, &b, &c);
sum = a+ b+ c;
printf("The sum of return 0;
}
```

OUTPUT:

```
Enter three numbers: 3 5 7
The sum of 3, 5, and 7 is: 15
```

EXERCISES I :

A.(Basic C Program Question)
Question 1: Write a C program to take two integers as input from the user and print their sum.

Question 2: Write a C program to find the square of a number entered by the user.

B.(Loop in C Program Question)
Question 1: Write a C program that uses a for loop to print the numbers from 1 to 10.

Question 2: Write a C program that uses a while loop to find the factorial of a number entered by the user.

Question 3: Write a C program that uses a do-while loop to print the multiplication table of a given number.

C.(One-Dimensional Array in C Program Question)
Question 1: Write a C program that initializes an array of 5 integers and prints the sum of all elements in the array.

Question 2: Write a C program to find the largest number in an array of 10 integers entered by the user.

Question 3: Write a C program to reverse the elements of an array of integers (with 5 elements) using a loop.

Question 4: Write a C program to input 5 numbers into an array, calculate their average, and print the result.