INTRODUCTION

As we know that C language is a procedural programming language. It is developed by dennis Ritchie the unix operating system at Bell labs in the year of 1972. C was invented to write an operating system called UNIX. C is a successor of B language which was introduce around the early 1970. The language was formalized in 1988 by the American National standard institude(ANSI). The UNIX OS was totally written in c.

Today C is the most widely used and popular system programming language**.** ‘C’ is a powerful programming language which is strongly associated with the UNIX operating system. Even most of the UNIX operating system is coded in ‘C’. Initially ‘C’ programming was limited to the UNIX operating system, but as it started spreading around the world, it became commercial, and many compilers were released for cross-platform systems. Today ‘C’ runs under a variety of operating systems and hardware platforms. As it started evolving many different versions of the language were released. At times it became difficult for the developers to keep up with the latest version as the systems were running under the older versions. To assure that ‘C’ language will remain standard, American National Standards Institute (ANSI) defined a commercial standard for ‘C’ language in 1989. Later, it was approved by the International Standards Organization (ISO) in 1990. ‘C’ programming language is also called as ‘ANSI C’.

Table of content

22. function for dynamic memory allocation in c.

23. Storage class in c.

24. void pointer in c.

25. Null pointer in c.

26. command line arguments in c.

27. Function pointer in c.

Variables & Data types in C

- A name given to a memory location
- Declared by writing type variable
- Initialized and declared by type variable

Data types in C

- Basic data types:- int , char, float, double
- Derived data types:- array, pointer, structures, union
- Enumeration data types:- emum
- Void data types:- void

Operator in C

An operator ia a symbol used to perform operation in given programming language. There are mainly five types of operator in C.

- Arithmetic operator (+, -, *, /)
- Relational operator (==, !=, *, /, %)
- Logical operator (&&, ||, !)
- Bitwise operator
- Assignment operator (=, +=, -=, *=, /=)

If else control statement in C

It is used to perform operation based on some conditions.

1. If statement

   Syntax:

   ```
   if(cond.)
   {
    Code
    }
   ```

2. If else statement

   Syntax:

   ```
   If(3>2)
   {
    Printf("This will execute");
    }
   ```

3. If else-if ladder

   Syntax:

   ```
   If(cond. 1)
   {
    Code 1
    }
    Else if(cond. 2)
    {
    Code 2
     }
      Else
      {
      Code n;
   ```

}

Switch case statement

1. Switch expression must be an int or char
2. Case value must be an integer or a character
3. Case must come inside switch
4. Break is not a must

Syntax for switch case

```
 Int a=2;
Switch (a)
{
Case 2:
Printf("value is 2");
Break;
Case 3:
Printf("value is 3")
Default:
Printf("Nothing matched");
}
```

Loops in C

Any repeated task more than one time is called loop.

Types of loop:

1. Do while loop:
    - Do while loop execute at least once
    - Do while loop is exit control loop
    - Do while loop is a post test loop
    - It test the condition at the end of the loop body
2. While loop:
    - While loop is entry control loop
    - A while loop is a pre test loop
    - It test the condition before executing the loop body
    - There is no semicolon at the end of the loop
3. For loop:
    - For loop execute the statement of program several times repeatedly unit a given condition returns false.

Break & Continue in c

- Break statement is used to terminates the loop.
- It is used in loop and switch case.
- Keyword break used in break statement.
- Control is transfer outside the loop.

Syntax:

Int i;

```
For(i=1;i<=10;i++)
{
If(i==5)
Break;
Printf("%d",i);
}
```

Continue:

1. Continue statement is used to continue the next iteration in the loop
2. It only used in loop
3. Keyword continue used in continue statement
4. Control remains in the same loop

Syntax:

```
Int I;
For(i=1;i<=10;i++)
{
If(i==5)
Continue;
Printf("%d",i)
}
```

Goto statement in c

- Also called jump statement in c
- Used to transfer program control to a pre defined label
- It uses it avoided since it cause confusion for the fellow programmer in understanding the code
- Goto statement is preferable when we need to break multiple using a single statement at the same time

Functions in c

- Function are used to divide a large c program into smaller pieces
- A function can be called multiple times to provide reusability and modularity to the c programming
- Also called procedure or subroutine

Syntax:

```
Int print star()
{
Printf("*");
Return 0;
}
```

Types of function:

1. Library function
2. Used defined function

Advantage of function:

1. We can avoid rewriting some logic through function
2. We can divide among programmers using function
3. We can easily a debug a program using function

Declaration, definition, & call

1. A function is declared to tell a compiler about its existence
2. A function is defined to get some task done
3. A function is called in order to be used

Recursion function in c

- Recursive function or recursion is a process when a function calls a copy of itself to work or a smaller problem.
- Any function which call itself is called recursive function
- This makes the life of program easy by dividing a given problem into easier
- A termination condition is imposed on such function to stop them executing copies of themselves for ever
- Any problem that can be solved recursively can also be solved iterately

Array in c

- An a is a collection of data items of the same type
- Iteration are stored at contiguous memory location
- It can also store the collection of derived data types such as pointer structures etc.
- A one-dimensional array is like a list
- A two dimensional array is like a table

- The c language place no limit on the number of dimensions in an array.

Types of array:

1. One-dimensional array
2. Two-dimensional array
3. Multi-dimensional array

Advantage of array:

- It is used to represent multiple date item of same type by suing only single name
- Accessing an item in a given array is very fast
- 2 dimensional array it easy in mathematical application as it is used to represent a matrix.

Pointer in C

- Variable which states the address of another variable
- Can be type int,char,array,function or any other pointer
- Size depend on architecture

  Ex: 2 bytes for 32 bit
- Pointer in c programming language can be declared (using*) asterisk symbol

Array and pointer arithmetic in c

There are four arithmetic operators that can be used on pointer.

- ++
- --
- +
- -

<u>**call by value & call by refrence in c**</u>

types of funcftion call in c programming:

in c programming language we can call a function in two different ways based on how we specify the argument are those two ways are:

1. Call by value:
   - When we call a function by value it means that we are passing the value of the arguments which are copied into the formal parameter of the function.
   - Which means that the original values remain unchanged and only the parameter inside the function changed.
2. Call by refrence:
   - The call by refrence method of passing arguments to a c function copies the address of the arguments into the formal parameters.
   - Address of actual arguments are copied and then assigned to the corresponding formal arguments.

Actual and formal parameters in c

- _When a function is called the value (expressions) that are passed in the call are called the arguments or actual parameters.

- Formal parameters are local variable which are assigned values from the arguments when the function is called.

String in C

- Array of character terminates by a null character.
- String in c is created by creating an array of characters.
- We need an extra character ('\0' or null character) to tell the compiler that the strings ends here.

Is string a data type in c

- No
- We have char, int, float, and other data types by no string data types in c.
- String is not a supported data type in c but it is a very useful concept used to model real world entities like name, city etc.
- We express string an array of character terminate by a null character('\0')

String.h library in c

- Strcat(): this function is used to concatenate or combine two given string.
- Strlen(): this function is used to show length of a string.
- Strrev(): this function is used to show reverse of a string.
- Strcpy(): this function is used to copy one string into another.

- Strcmp(): this function is used to compare two given string.

Structures in c

- Structures are user defined data types in c.
- Using structures allows us to combine data of different types together.
- It is used to create a complex data type which contain diverse information.
- They are very similar to array but structure can store data of any type which is practically more useful.

Unions in c

- Union is a user defined data types (very similar to structures)
- The difference between structures and union lies in the fact that in structure each member has its own storage location where as member of a union user a single shared memory location.
- This single shared memory location is equal to the size of its largest data member.

Declaring & accessing union member in c

- Like structure e access any member by using the member access operator() in union.
- We use keyword union to define a union.
- Syntax is very similar to that of structure

Static variable in c

- Static variable are variable which have a property of preserving their value even when they go out of scope.
- They preserve their value from the previous scope and are not initialized again.
- Static variable remains in memory through out the spam of the program.
- Static variable are initialized to 0 if not iniyialized explicitly.
- In c, static variable can only be initialized using constant literals.

Local variable(RECAP)

- Scope is a region of the program where a defined variable can exit and beyond which is cannot be accessed.
- Variables which are accessed inside a function or block are called local variable.
- They can only be accessed by the function they are declared in!
- They are inaccessible to the function outside the function they are declared in!

Global variable(RECAP)

- These are the variable defined outside the main method .
- Global variable are accessible through the centre program from any function.
- If a local and global variable has the same name, the local variable will take preference.

Formal arguments:

- These variables are treated aslocal variable with in a function
- These variable take precedence over global variable.

Dynamic memory allocation in c

- An statically allocated variable or array has a fixed size in memory
- We have learned to create big enough array to fit in our input but this doesn’t seem like an optional way to allocated memory.
- Memory is very useful resource
- Clearly we need a way to request memory on run time
- Dynamic memory allocation is a way in which the size of a data structure can be hanged during the run time.

Static memry allocation in c

- Allocation is done before the program execution.
- There is no memory resusability and the memory allocated cannot be freed.
- Less efficient

Memory allocation in c

Memory assigned to a program to a program in a typical architecture can be broken down into four segment.

1. Code
2. Static/global variable
3. Stack
4. Heap

Function for dynamic memory allocation in c

- In dynamic memory allocation the memory is allocated at run time from the heap segment

- We have four function that heap us achieve this task:
  1. Malloc
  2. Calloc
  3. Realloc
  4. Free

Malloc():

- malloc stands for memory allocation.
- It reserve a block of memory with the given amount of bytes.
- The return value is a void pointer to the allocated space.
- All the values at allocated memory are initialized to garbage value.

Calloc():

- Calloc() stands for contiguous allocation.
- It reserves n blocks of memory with the given amount of bytes.
- The return value is a void pointer to the allocated space.
- All the values atallocated memory are initialized to 0.

Realloc():

- Realloc() stands for reallocation.
- If the dynamically allocated memory is insufficient we can change the size of previously allocated memory using realloc(() function.

<u>Storage class in c</u>

- A storage class defined scop default initial value & life time of a variable.

- In previous lecture we show the dynamic memory allocation ia a way in which the size of data structure can be changed during the run time.
- Memory assigned to a program in a typical architecture can be broken down into four segment.
  1. Code
  2. Static/global variable
  3. Stack
  4. Heap

Automatic variable: auto storage class:

- Scope: local to the function body they are totally defines in.
- Default value: garbage value a London value.
- Life time: till the end of the function block they are defined in.

External variable: external storage class:

- They are same as global variable.
- Scope: global to the program they are defined in.
- Default initial value: 0

Register variable: register storage class:

- Scope: local to the function they are defined in.
- Default initial value: garbage value.
- Life time: they are available till thye end of the function block in which the variable is defined.

<u>**Void pointer in c**</u>

- A void pointer is a pointer that has no data type associated with it.
- A void pointer can be easily type castedto any pointer type.
- In simple language it is generally purpose pointer variable.

Uses of void pointer

- In dynamic memory allocation malloc() & calloc() return(void*) type pointer.
- This allows there dynamic memory function to be used to allocated memory of any data type. This is because these pointer can be type casted to any pointer type.

Null pointer in c

- Null pointer is a pointer which has value reserved for indicating that the pointer or refrence does not refer to a valid object
- A null pointer is guaranteed to compare unequal to any pointer that points to a valid object.
- Dereferencing a null pointer is undefined behaviour in c and a conforming implementation is allowed to assume that any pointer that is dereferenced is not a null.

Uses of null pointer

- To check for legitimate address location before accessing any pointer variable.
- By doing so we can perform error handling while using pointer with c.
- Example of such error handling can be dereference pointer variable only if it not null.

Command line argument in c

- Command line argument in an important concept in c programming.
- Sometimes we need to pass arguments from the command line to the program a set of input.
- Command line arguments are used to supply parameter to the program hen it is invoked.
- These arguments are passed to the main() method.

Function pointer in c

- We can have pointer pointing to function as well.
- Function pointer point to code and not data.
- Function pointer are useful to implement call back function.

Call back function

- Function pointer are used to pass a function to a function.
- This passed function can then be called again.
- This provide programming to write less code to do more stuff.

CONCLUSION

We originally set out to explore the strengths and weaknesses of five of the most commonly used programming languages. This exploration was guided by the aim to be able to make more informed decisions of which programming language might be best suited for a particular situation. We have looked at C, Java, JavaScript, Python, and C++ over the course of the past few weeks, and today we will recap what has been learned about what types of application are suited for each language. C was the first programming language we looked at.

C is most useful for embedded systems, or applications that require the ability to be light-weight and have precise control over system resources. C is lacking a lot of the functionality that more contemporary languages feature, but remains a core tool for Unix developers. When we set out to build an embedded system, we should first consider using C.

- Great portability and deterministic usage.
- Versability efficiency, and good performance.
- World is running on c-powered device.
- C is the past, the present and far we can see still the future for many areas of software.