STORAGE CLASSSES

Declaration of variables contains two part: -

Data type of the variable it tells the type of information represented by a variable, e.g., integer number, floating number, character etc

Storage class refers to the portion of the program over which the variable is recognized. There are basically two types of location **Memory and CPU register** and the variable of the storage class determine the location of the stored value.

Storage class of the variable defines

- Place where the variable should store
- Default or initial value of the variable
- Scope of the variable, the function where the value of variable is available
- Life of the variable how long it exists

Storage classes in C 4 types

- Automatic storage class.
- register storage class
- Static storage class.
- external storage class

Automatic storage class

Synatax: **-** auto int x;

In here it exist only under the variable in which it is declared. The features of such storage class are

- **Storage:** Memory
- **Default initial value:** An unpredictable value, garbage value
- **Scope:** local to the variable in which it is declare
- **Life:** Control remains within the block in which it is declared

e.g., for(int i=1;i<6;i++)

{ printf(“hello world);

Printf(“%d”,i);

In the above program an error that I is not declared will be out because it is not declared outside the function for.

Register storage class

***Syntax:* -** register int x;

It is used in the case where we need to use register memory. These register memories are special storage unit inside the CPU and basically the arithmetic and logical operation comprising a programme are carries out here. These are basically the used to execute data faster and it should be used limited as it may be used by other applications within the computer and are found very less in the CPU.

- **Storage:** CPU registers
- **Default intial value:** garbage value
- **Scope:** local to which the variable is declared
- **Life:** Till the control is defined

Static storage class

***Syntax:* -** static int x;

It's a single file programme and the value of the static variable exist throughout the programme. Its value gets stored with the conditions its called up and throughout the programme. It is used when we want the value to be persist between different function call

- **Storage: -** Memory
- **Default initial value: -** zero
- **Scope: -** local to the function where it is declared
- **Life: -** Value of variable persist between different function call.

e.g., main()

{ display();

Display();

Void display()

{

Static int c=1

C +=5;

Printf("%d ",c)

}

Output

6 11

We can see that the first function call makes the value of c to 6 and in the next call the changed value 6 is used that is this call changes the value of variable declared

External storage class

It is written outside all the function and it is also known as global variable. And any storage class specifiers is not used. Should be used in the case where it is urgent. And unnecessary declaring variable as global may result in wastage of storage space.

- **Storage: -** memory
- **Default initial value: -** zero
- **scope: -** global
- **life: -** as along the program as execution doesn't come to end

e.g., int n=5;// global variable

int main()

{ ++n;

Display();

}

Int display()

++n;

Printf("n=%d",n);

}

Output

n = 7

Storage classes	**Storage**	**Default initial value**	**Scope**	**Life**
Automatic	memory	Garbage value	Local to the function or block	Till the function is terminated
Register	CPU register	Garbage value	Same as above	Same as above
Static	Memory	Zero	Same as abo0ve	Value passes between different function calls
External	Memory	Zero	Global	As long as the program's execution ends

Important questions

1. **What does the storage class of a variable refer to?**
   Storage class refers to the portion of the program over which the variable is recognized. There are basically two types of location **Memory and CPU register** and the variable of the storage class determine the location of the stored value.
2. **What does storage class of a variable define?**
   - Place where the variable should store
   - Default or initial value of the variable
   - Scope of the variable, the function where the value of variable is available
   - Life of the variable how long it exists

3. **Where is the values of variables defined inside a program stored?**
   Basically, there are 2 locations where the variable declared in a storage can be stored memory and CPU registry.

4. **Name the storage classes used in C.**
   - Automatic storage class.
   - register storage class
   - Static storage class.
   - external storage class

5. **Write down the syntax for a)Automatic storage class b)Register storage class c) Static storage class d)External storage class**
   a, auto int x;
   b, register int x;
   c, static int x;
   d, int n=5; // declared outside the programme
6. **What is the default storage class?**
   Default storage classes are the storage class assumption made by the compiler when we don't make any specifications.
7. **Write down the features of variables defined inside a)Automatic storage class b)Register storage class c) Static storage class d)External storage class**

Storage classes	**Storage**	**Default initial value**	**Scope**	**Life**
Automatic	memory	Garbage value	Local to the function or block	Till the function is terminated
Register	CPU register	Garbage value	Same as above	Same as above
Static	Memory	Zero	Same as abo0ve	Value passes between different function calls
External	Memory	Zero	Global	As long as the program's execution ends

8. **What is the advantage of declaring a variable as a register variable?**
   These are faster than local variables these are special unit present in the CPU. And the arithmetic and logical operations inside a programme is carried out here.
9. **Which type of variables are to be declared inside register storage class?**
   only the variables that are used very often in a program should be declared in register class as there are only few CPU registers available.
10. **Applications of register storage class variables**
    Loop counters because it get used a number of times in a program.

11. **What is the difference between auto and static variables?**

    The value of auto variable exist only in the function block and its default initial value is garbage value, While in the case of static variable the value exist throughout the programme

12. **Which type of variables are to be declared as extern? Why?**

    The variables that need to be accessed throughout the programme within all the functions

13. **What is the disadvantages of using external variables?**
    Declaring all the variables as extern will result in the wastage of memory space as they remain active throughout the programme

14. **What is the output of the following program?**

    #include int main()

    { static int a = 3;

    printf("%d", a --);

    return 0; }

    a) 0

    b) 1

    c) 2

    d) 3

    the answer is 3

15. **What will be the output of the following program?Justify your answer.**

    #include

    void main()

    { fun();

    Return 0; }

    void fun()

    {

auto int I = 1;

register char a = ‘D’;

static int p = 0;

printf(“%d %d %ld”, I, a, p);

}

a) 1 D 0

b) 1 0 0

c) 0 D 1

d)1 68 0

The answer is 1 68 0 as in this case a is taken as character and the integer value of ‘D’ is asked to print which will print the ASCII value all othe variables they are integer variables.

16. What will be the output of the program? Justify your answer.

extern int i = 5;

main()

{ int i=7;

printf("%d",i);

}

A. 5

B. compiler error

C. 7

D. garbage value

7 is the answer because a value is declared inside the function and the printf is inside that function of the value declared. 5 will be displayed only if the I is not initialised and simply called in the function.