Course Name: O Level (2nd Sem B1 Batch) **Subject:** C Language
Topic: Storage Class in C **Date:** 17-April-2020

Storage Class in C

'Storage' refers to the scope of a variable and memory allocated by compiler to store that variable. Scope of a variable is the boundary within which a variable can be used. Storage class defines the scope and lifetime of a variable.

Functions of storage class:

- To determine the location of a variable where it is stored?
- Set initial value of a variable or if not specified then setting it to default value.
- Defining scope of a variable.
- To determine the life of a variable.

Types of Storage Classes:

Storage classes are categorized in 4 (four) types as,

- Automatic storage class
- Register storage class
- Static storage class
- External storage class

Automatic Storage Class:

Automatic variables are declared inside a function in which they are to be utilized. They are created when the function is called and destroyed automatically when the function is exited, hence the name automatic. Automatic variables are therefore private to the function in which they are declared. Because of this property, automatic variables are also referred to as local or internal variable.

Keyword	:	**auto**
Storage Location	:	**Main memory**
Initial Value	:	**Garbage Value**
Life	:	**Control remains in a block where it is defined.**
Scope	:	**Local to the block in which variable is declared.**

Syntax : auto [data_type] [variable_name];
Example : auto int a;

Example:

```
#include <stdio.h>
void main( )
{
  auto int i = 1;
  {
    auto int i = 2;
    {
      auto int i = 3;
      printf ( "\n%d ", i);
    }
    printf ( "%d ", i);
  }
  printf( "%d\n", i);
}
```

OUTPUT

3 2 1

In above example program you see three definitions for variable i. Here, you may be thinking if there could be more than one variable with the same name. Yes, there could be if these variables are defined in different blocks. So, there will be no error here and the program will compile and execute successfully. The printf in the inner most block will print 3 and the variable i defined in the inner most block gets destroyed as soon as control exits from the block. Now control comes to the second outer block and prints 2 then comes to the outer block and prints 1. Here, note that automatic variables must always be initialized properly, otherwise you are likely to get unexpected results because automatic variables are not given any initial value by the compiler.

Register Storage Class:

We can tell the compiler that a variable should be kept in one of the machine's register instead of keeping in the memory where normal variables are stored. Since a register access is much faster than a memory access. Keeping frequently accessed variables in the register will lead to faster execution of program.

Keyword	:	**register**
Storage Location	:	**CPU Register**
Initial Value	:	**Garbage**
Life	:	**Local to the block in which variable is declared.**
Scope	:	**Local to the block.**

Syntax **:** register [data_type] [variable_name];
Example **:** register int a;

Try Yourself:

1. What is storage class?
2. How many types of storage class in C.
3. What is internal variable?
4. What is the difference between local and the register storage class?