

STORAGE CLASSES IN C++

BY: NIKHIL KUMAR

STORAGE CLASS



- The storage class determines the part of the memory where the variable would be stored.
- The storage class also determines the initial value of the variable.
- and it used to define the scope and lifetime of variable.
- There are two storage location in computer :
CPU Registers and Memory

CPU REGISTER AND MEMORY



- A value stored in a CPU register can always be accessed faster than the one that is stored in memory.

TYPES OF STORAGE CLASSES



There are four types of storage classes in C:

- i. Automatic storage class
- ii. Register storage class
- iii. Static storage class
- iv. External storage class

Storage Class	Keyword	Lifetime	Visibility
Automatic	auto	Function Block	Local
External	extern	Whole Program	Global
Static	static	Whole Program	Local
Register	register	Function Block	Local

Automatic Storage Class



- Keywords : auto.
- Storage : memory.
- Default initial value : garbage value.
- Scope : local to the block in which the variable is defined.
- Life : till the control remains within the block in which the variable is defined.

Automatic Storage Class

Syntax of Automatic Storage :

```
datatype var_name1 [= value];
```

Or

```
auto datatype var_name1 [= value];
```

Example of Automatic Storage Class :

```
auto int x;  
float y = 5.67;
```

Example of Automatic Storage Class



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

Output:

3 2 1

Register Storage Class



- Keywords : register.
- Storage : CPU Register.
- Default initial value : garbage value.
- Scope : local to the block in which the variable is defined.
- Life : till the control remains within the block in which the variable is defined.

Register Storage Class

Syntax of Register Storage Class

```
register datatype var_name1 [= value];
```

For example

```
register int id;  
register char a;
```

Example of Register Storage Class

```
#include<stdio.h>
#include<conio.h>
void main()
{
register int i;
for(i=1;i<=10;i++)
printf(" %d",i);
getch();
}
```

Output:

1 2 3 4 5 6 7 8 9 10

Register Storage Class



- If the microprocessor has 16-bit registers then they cannot hold a float value or a double value which requires 4bytes(32-bit) and 8 bytes(64-bit)
- If you want to use the register storage class(16-bit microprocessor) with float and double variable then you won't get any error messages. Your compiler would treat the variables as auto storage class.

Static Storage Class



- Keywords : static.
- Storage : memory.
- Default initial value : zero.
- Scope : local to the block in which the variable is defined.
- Life : value of the variable persists between different function calls.

Static Storage Class



Syntax of Static Storage Class

```
static datatype var_name1 [= value];
```

For example

```
static int x = 101;  
static float sum;
```

Dif. b/w auto and static storage class

Automatic

```
#include<stdio.h>
#include<conio.h>
increment();
void main()
{
increment();
increment();
increment();
}
increment()
{
auto int i=1;
printf("%d\t",i);
i++;
getch();
}
Output:
1 1 1
```

Static

```
#include<stdio.h>
#include<conio.h>
increment();
void main()
{
increment();
increment();
increment();
}
increment()
{
static int i=1;
printf("%d\t",i);
i++;
getch();
}
Output:
1 2 3
```

External Storage Class



- Keywords : extern.
- Storage : memory.
- Default initial value : zero.
- Scope : global.
- Life : as long as the program's execution doesn't come to an end.

External Storage Class

Syntax of External Storage Class :

```
extern datatype var_name1;
```

For example:

```
extern float var1;
```

- The difference b/w two programs 1st auto and 2nd static storage class for variable 'i' the scope of auto and static both use local to the block in which the variable is declared.
- Those program consists two functions main() and increment().
- The increment() function called from main() function for three times.
- Each time increment the value of 'i' and print.
- when variable 'i' is auto each time increment and re-initialized to 1.

Example of External Storage Class

```
#include<stdio.h>
#include<conio.h>
int i =1;
increment();
void main()
{
printf("%d\t",i);
increment();
increment();
getch();
}
increment()
{
i++;
printf("%d\t",i);
}
Output:
1 2 3
```



Thank You!