

08-06-2024

## STORAGE CLASSES =>

1. Auto
2. Register
3. Static
4. Extern

### 1. AUTO :-

- > storage location = RAM [CPU registers]
- > default initial value = garbage
- > scope = local to block
- > life time = control within the block

### Ex:-

-> main()

```
{  
    auto int a = 5;  
    {  
        auto int a = 6;  
        {  
            auto int a = 7;  
            printf("%d", a);  
        }  
        printf("%d", a);  
    }  
    printf("%d", a);  
}
```



## 2. REGISTER :-

- Storage location = CPU registers
- default initial value = garbage
- scope = Local to block
- Life time = Control within the block

Ex :-

```
→ main ( )
```

```
{
```

```
    register int a = 5;
```

```
    {
```

```
        register int a = 6;
```

```
        {
```

```
            register int a = 7;
```

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

```
        }
```

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

```
    }
```

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

```
}
```

### 3. STATIC :-

- main  $\rightarrow$  Storage location = RAM
- Default initial value = zero
- Scope = Local to block
- Life Time = end of the program

Ex:-

→ main ( )

{

    incr ( );

    incr ( );

    incr ( );

}

incr ( )

{

    static int i = 1;

    printf ("%d", i);

    i++;

}



#### 4. EXTERN:-

→ Storage location = RAM

→ Default initial value = zero

→ Scope = global

→ Life time = end of the program

Ex:-

→ main ( )

{

extern int a, b, c;

c = a + b;

printf ("%d", c);

}

int a = 5, b = 6, c;

→ int a = 5, b = 6, c;

main ( )

{

add ();

}

add ( )

{

c = a + b;

printf ("%d", c);

}