Computer Programming & Problem Solving

CS100

Mrs Sanga G. Chaki
Department of Computer Science and Engineering
National Institute of Technology, Goa

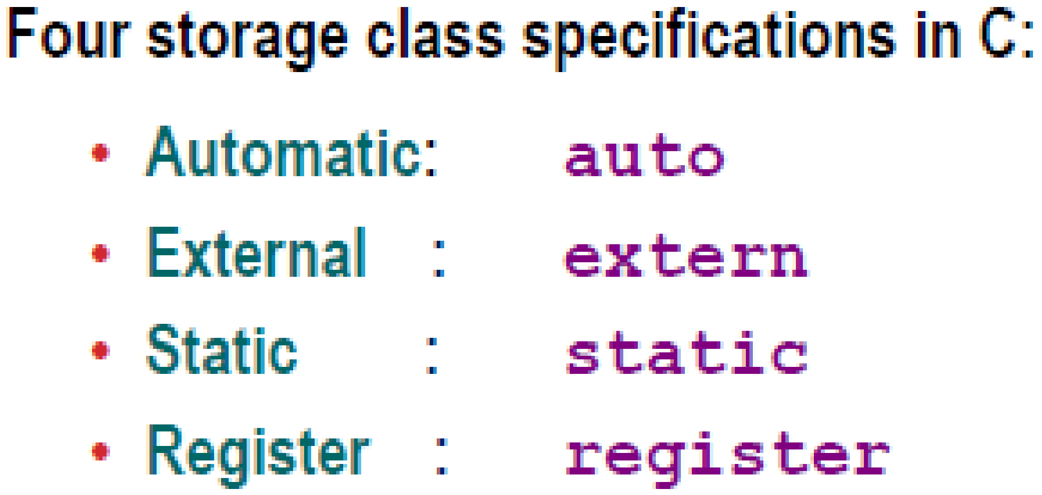
May, 2023

Storage Classes

What are Storage Classes in C?

1. **A variable's storage class tells us**
   a) **Where the variable would be stored.**
   b) **What will be the initial value of the variable, if initial value is not specifically assigned**
   c) **What is the scope of the variable**
   d) **What is the life of the variable**
2. **Till now why we have not mentioned storage classes?**
   a) **Because storage classes have defaults**
   b) **The compiler will assume a storage class depending on the context in which the variable is used.**

Types of Storage Classes in C



Four storage class specifications in C:

- Automatic: `auto`
- External : `extern`
- Static : `static`
- Register : `register`

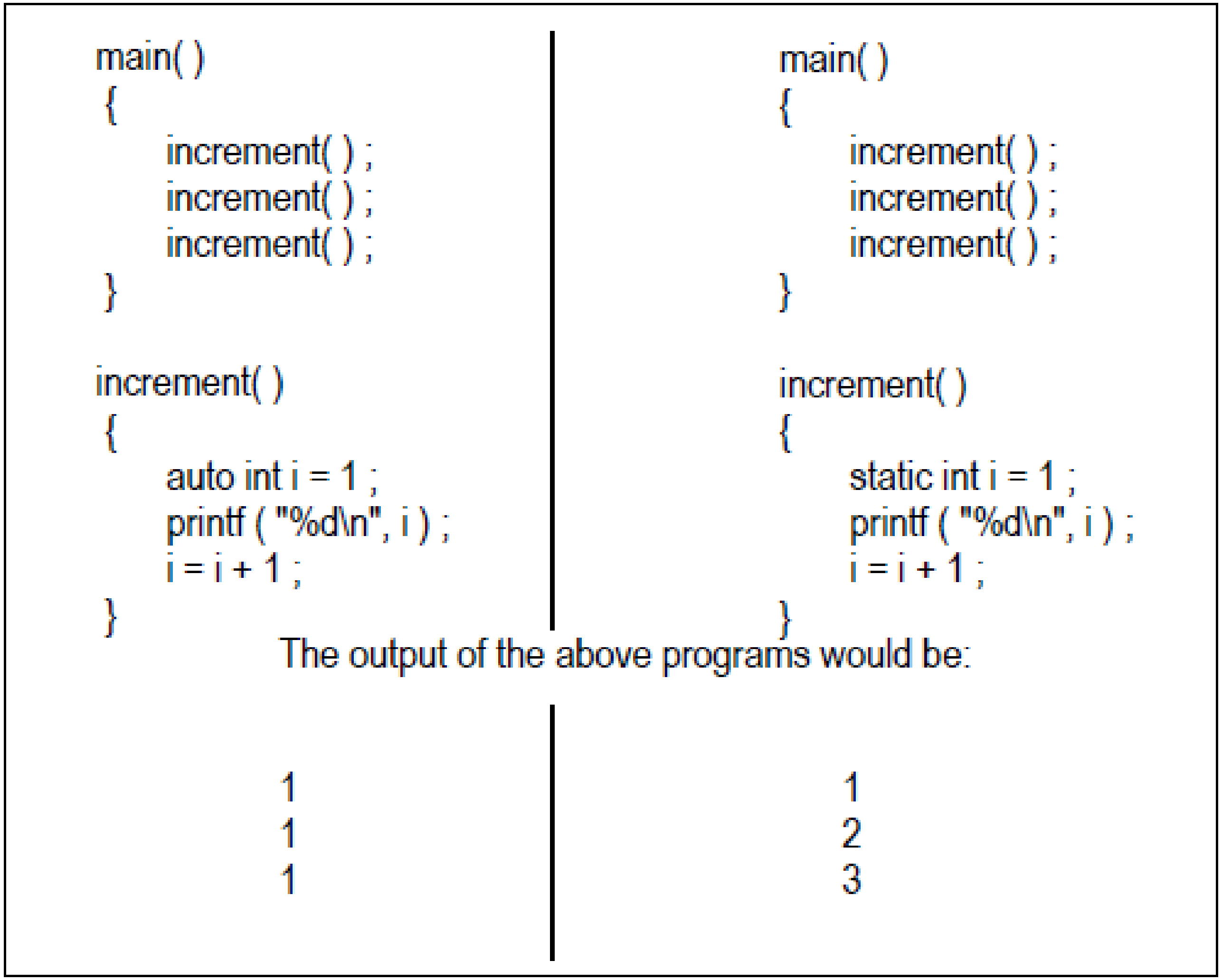
Automatic Variable

1. **These are always declared within a function and are local to the function in which they are declared.**
2. **This is the default storage class specification.**
3. **The keyword auto is optional.**
4. **Stored in memory**

Static Variables

1. **Are defined within individual functions and have the same scope as automatic variables.**
2. **Unlike automatic variables, static variables retain their values throughout the life of the program.**
3. **Stored in memory**

Automatic vs Static Variables



```
main( )
{
    increment( ) ;
    increment( ) ;
    increment( ) ;
}

increment( )
{
    auto int i = 1 ;
    printf ( "%d\n", i ) ;
    i = i + 1 ;
}
```

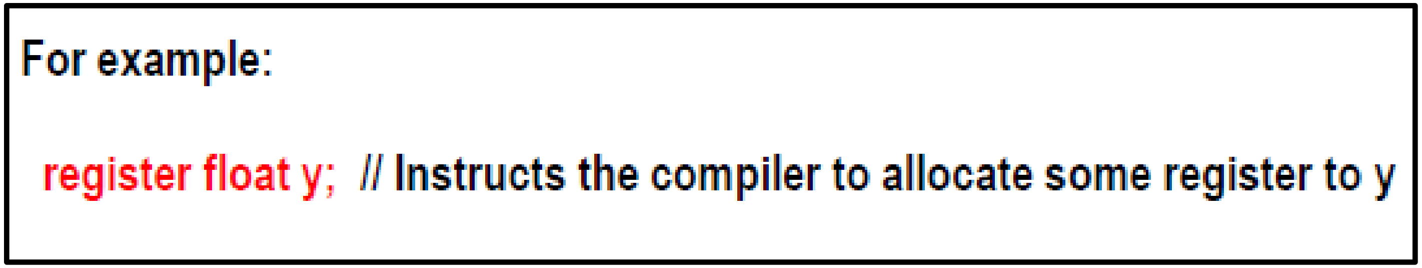
```
main( )
{
    increment( ) ;
    increment( ) ;
    increment( ) ;
}

increment( )
{
    static int i = 1 ;
    printf ( "%d\n", i ) ;
    i = i + 1 ;
}
```

The output of the above programs would be:

```
1          1
1          2
1          3
```

Register Variables

1. **These variables are stored in high-speed registers within the CPU.**
2. **Commonly used variables may be declared as register variables.**
3. **Results in increase in execution speed.**



For example:

```
register float y;  // Instructs the compiler to allocate some register to y
```

External Variables

1. **They are not confined to a single file. They are global variables.**
2. **Global variables – declared outside all functions (even main).**
3. **Their scope extends from the point of definition through the remainder of the program**

When to use which?

1. **Use static storage class only if you want the value of a variable to persist between different function calls.**
2. **Use register storage class for only those variables that are being used very often in a program.**
3. **Use extern storage class for only those variables that are being used by almost all the functions in the program**
4. **Use auto storage class if none of the above needs are present.**