C programming

Variable in C

Lesson Objectives

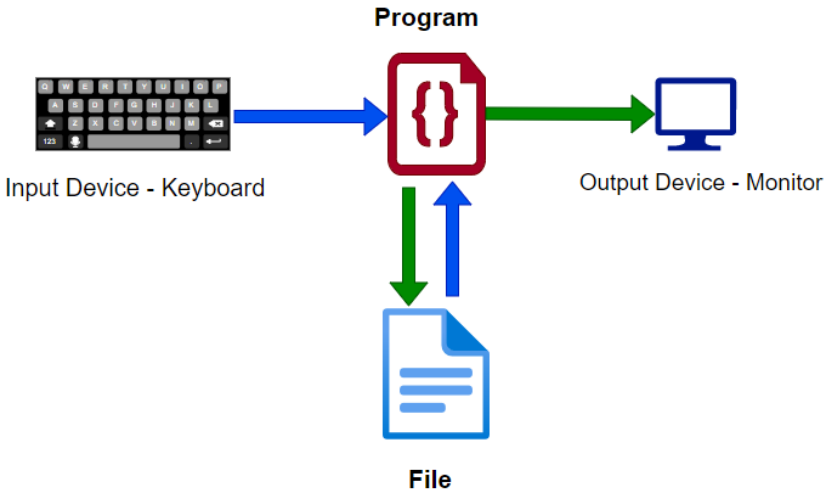
- Introductory question
- Basic Data Types
- Store Class
- Key words for variable
- Pointer variable
- Structure data type

Section 1

INTRODUCTORY QUESTION

Introductory question

- ✓ ***What is variable?***
- ✓ ***Why do need variable in programming?***



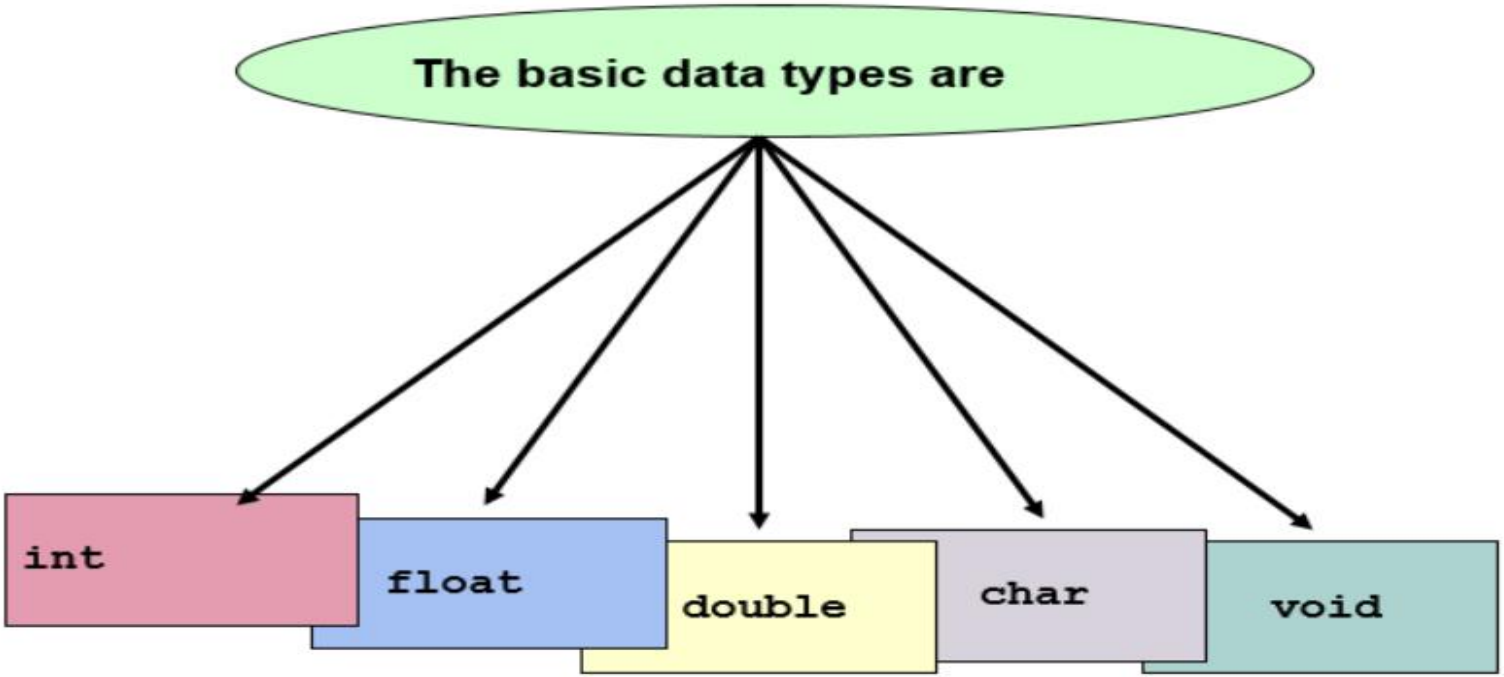
Section 2

BASIC DATA TYPES

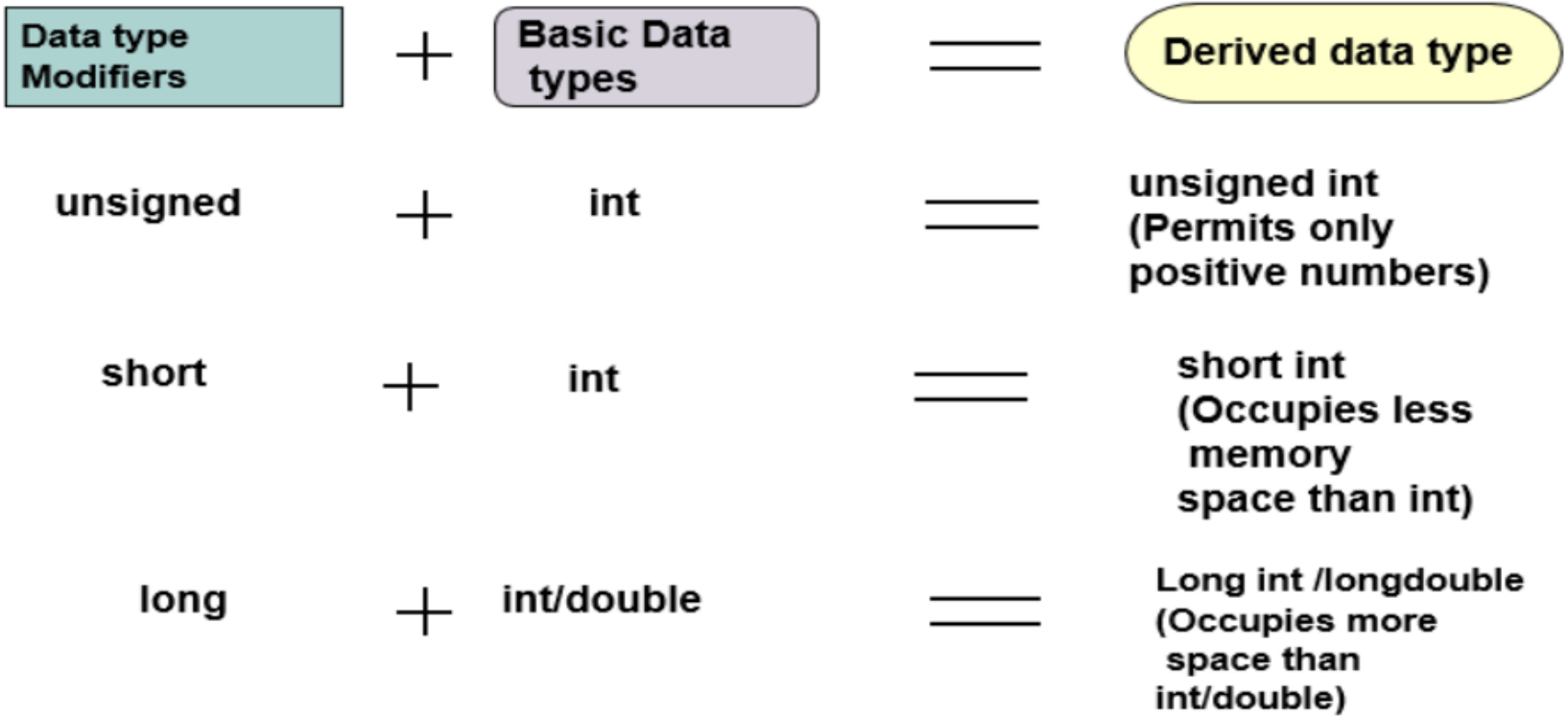
Basic Data Types. Agenda

- *Basic data types in C*
- *Derived Data Types*

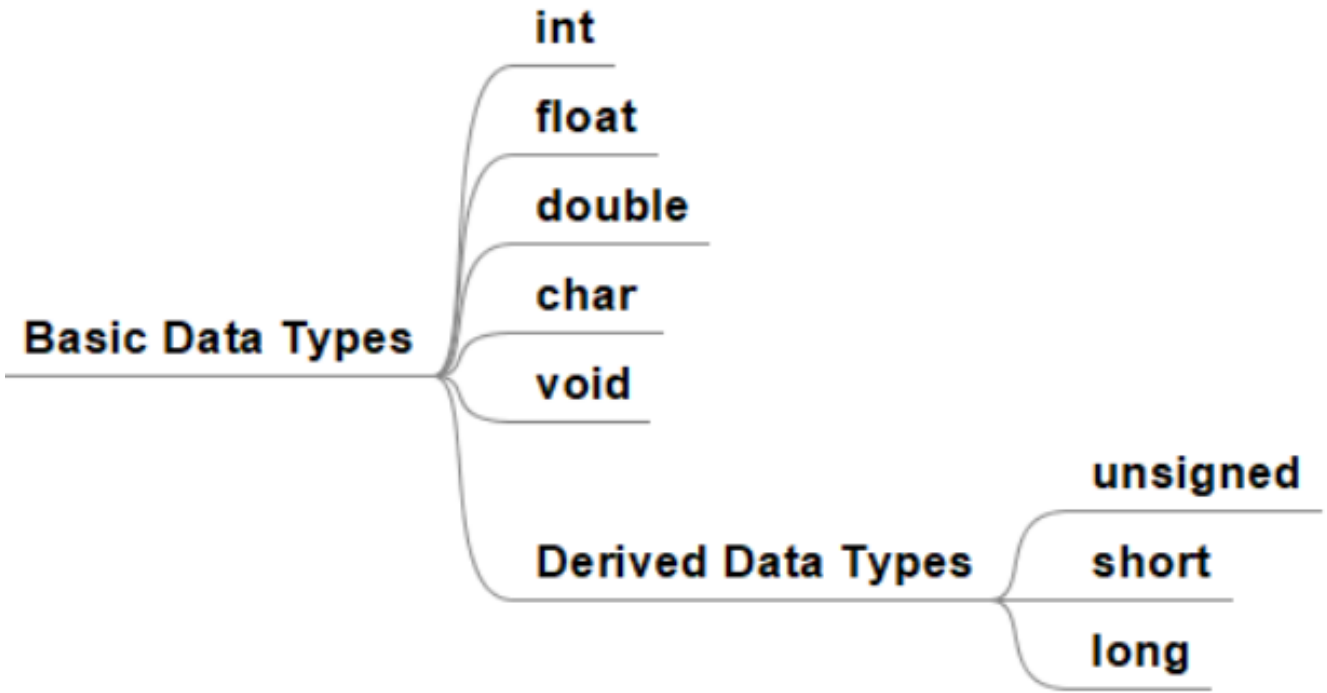
Basic Data Types in C



Derived Data Types



Basic Data Types. Summary



Section 3

STORE CLASS

Store Class. Agenda

- *Time life of a variable*
- *Scope of a variable*
- *Variable in memory*
- *Global and Local discriminative*

Time life of a variable

- *When was a variable allocated in memory?*
- *When was a variable destroyed?*

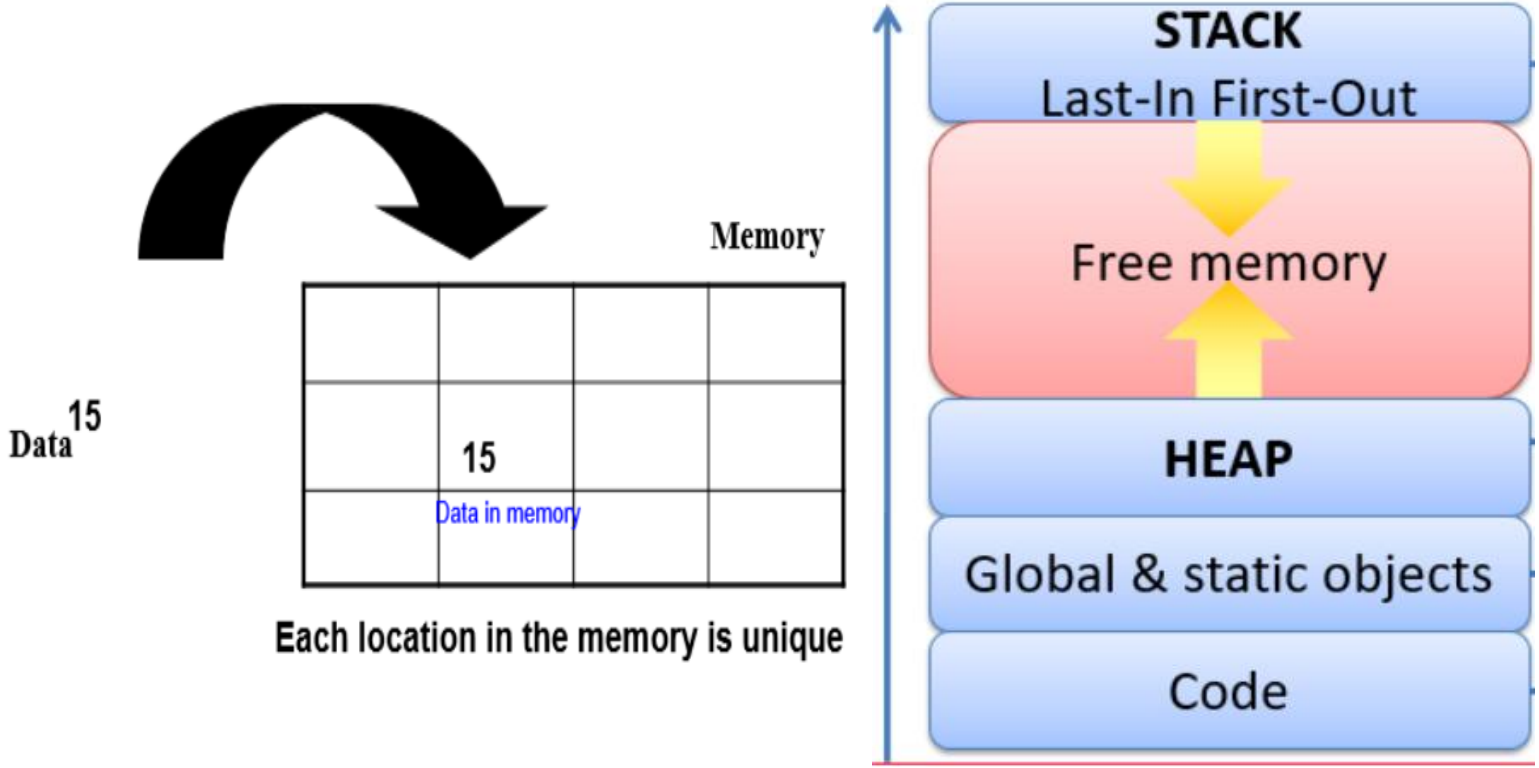
The **lifetime** of a variable is the time period in which the variable has valid memory.

Scope of a variable

- *Where was a variable declared in the program?*

The **scope** of a declaration is the part of the program for which the declaration is in effect.

Variable in memory

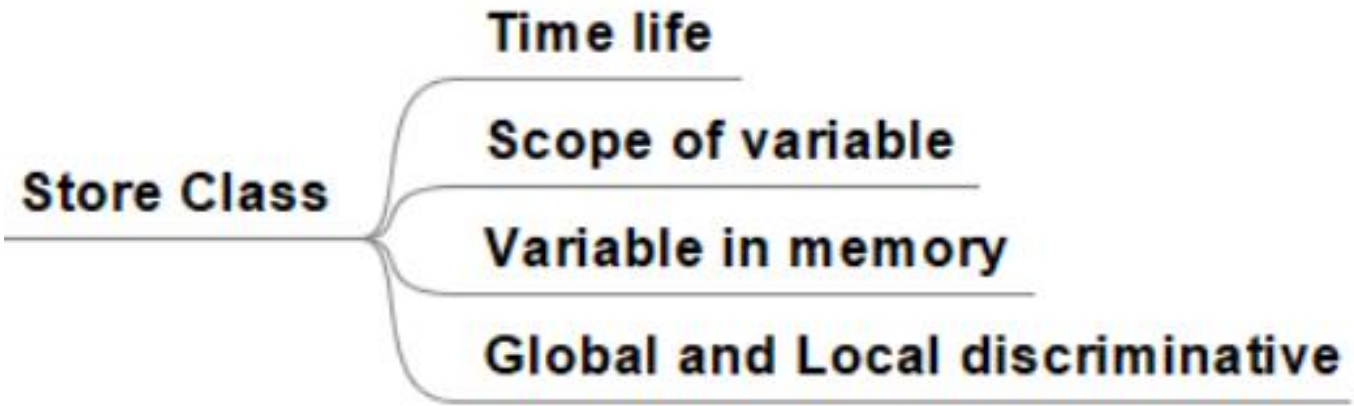


Global and Local variable discriminative

- Global variables are declared outside any function, and they can be accessed (used) on any function in the program. Local variables are declared inside a function, and can be used only inside that function. It is possible to have local variables with the same name in different functions.

- Time life ?
- *Scope ?*
- *Memory ?*

Store class. Summary



Section 4

KEY WORD FOR VARIABLE

Key word for variable. Agenda

- *static*
- *extern*
- *register*
- *volatile*

Key word: static

- In some programming languages such as C (and its close descendants like C++, Objective-C, and Java), static is a reserved word controlling both lifetime (as a static variable) and visibility (depending on linkage). The effect of the keyword varies depending on the details of the specific programming language.
  (wiki)
- Static global variable ?
- Static function ?
- Static local variables ?

Key word: extern

- **Extern** is a keyword in C programming language which is used to declare a global variable that is a variable without any memory assigned to it. It is used to declare variables and functions in header files. Extern can be used access variables across C files.
- **Extern** variable ?
- **Extern** function ?

Key word: register

- Register variables tell the compiler to store the variable in CPU register instead of memory. Frequently used variables are kept in registers and they have faster accessibility. We can never get the addresses of these variables. “register” keyword is used to declare the register variables.

(wikis)

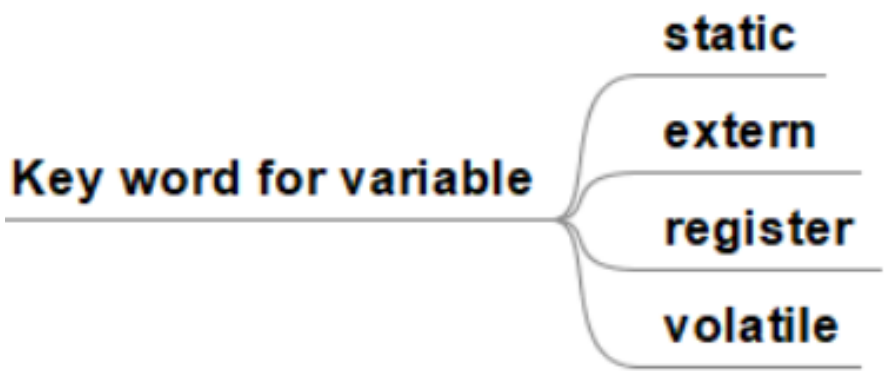
Key word: volatile

- The volatile keyword is intended to prevent the compiler from applying any optimizations on objects that can change in ways that cannot be determined by the compiler.

User case

- *What is the key words: static,…?*
- *When will use static, extern, register, volatile key words?*

Key word for variable. Summary



Section 5

POINTER VARIABLE

Pointer variable. Agenda

- *What is pointer?*
- *User case?*

What is pointer?

- A **pointer** is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before using it to store any variable address.

Int a;

a = 5;

User case?

We have 2 lesson about user case for pointer.

Pointer variable. Summary

- *Understanding about pointer variable.*

Section 6

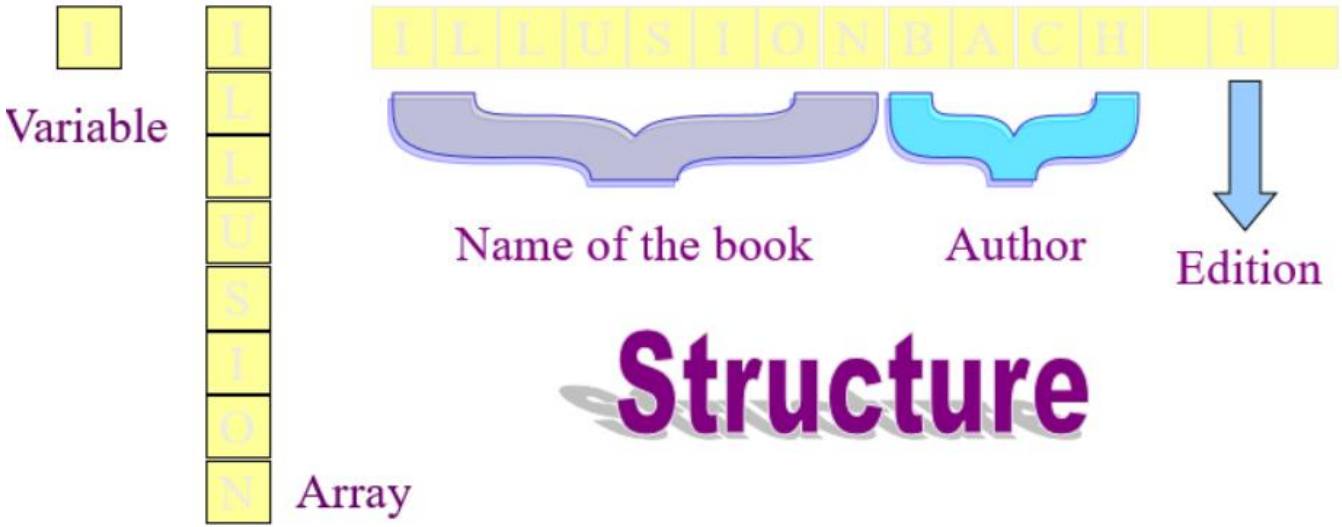
STRUCTURE DATA TYPE

Structure data type. Agenda

- *Structure variable*
- *Union*
- *Enum*

What is structure data type?

- A structure consists of a number of data items, which need not be of the same data type, grouped together
- The structure could hold as many of these items as desired



Define a structure data type.

- A structure definition forms a template for creating structure variables
- The variables in the structure are called structure elements or structure members
- Example:

```
struct cat
   {  char bk_name [25];
      char author [20];
      int edn;
      float price;
     };
```

Declare structure data type variables.

- Once the structure has been defined, one or more variables of that type can be declared
- Example: **struct cat books1;**
- The statement sets aside enough memory to hold all items in the structure

Other ways

```
struct cat {    char bk_name[25];
                char author[20];
                int edn;
                float price;
        } books1, books2;
```

```
struct cat books1, books2;
```

or

```
struct cat books1;
struct cat books2;
```

Structure data type variables.

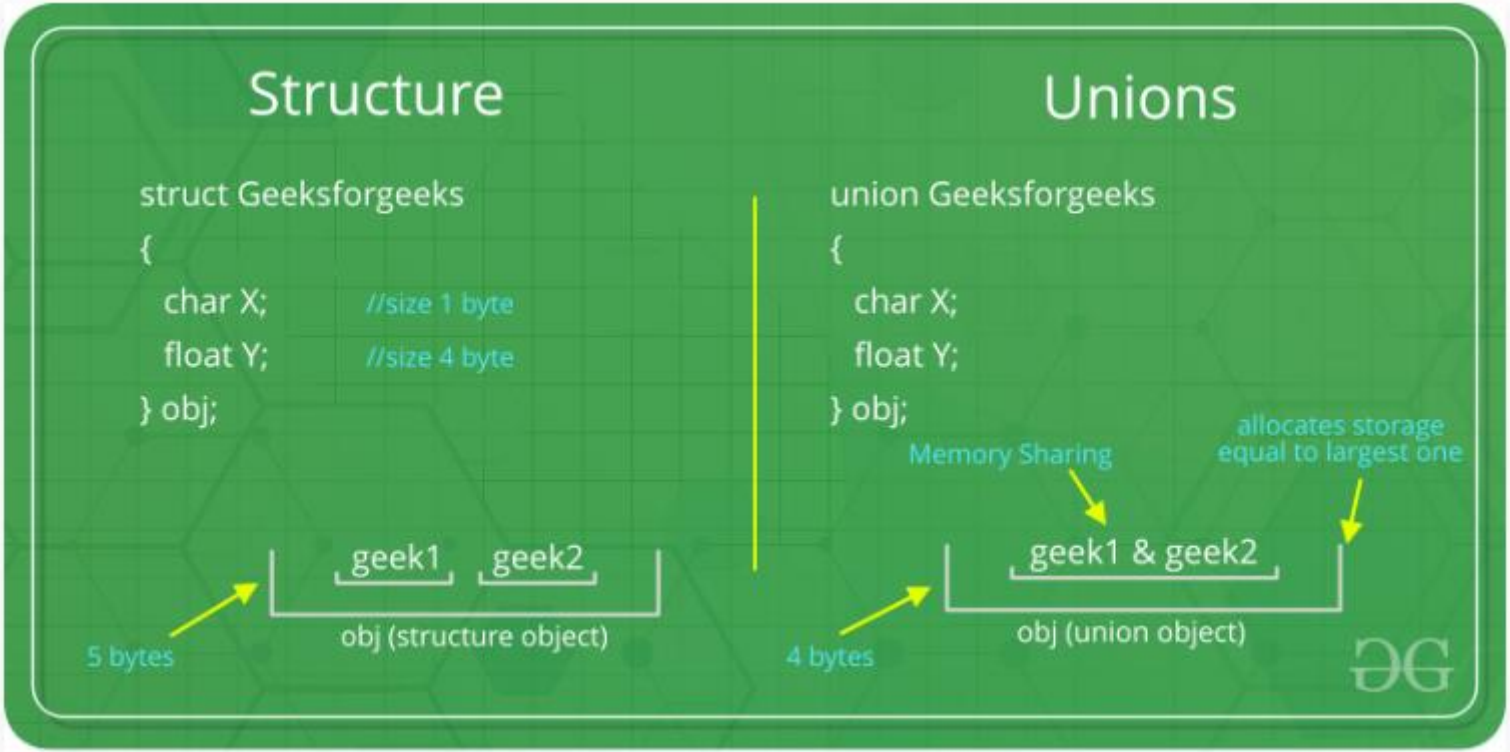
- Size of structure data type ?
- Data structure alignment ?
- User case ?
- What is typedef?

Union variable

- What is Union?
- Define Union.
- Declare Union variables.
- Size of Union?
- User case?

Union variable: What is Union?

A **union** is a special data type available in C that allows to store different data types in the same memory location. You can define a union with many members, but only one member can contain a value at any given time. Unions provide an efficient way of using the same memory location for multiple-purpose.


Structure
struct Geeksforgeeks
{
char X; //size 1 byte
float Y; //size 4 byte
} obj;
geek1 geek2
obj (structure object)
5 bytes
Unions
union Geeksforgeeks
{
char X;
float Y;
} obj;
Memory Sharing
allocates storage equal to largest one
geek1 & geek2
obj (union object)
4 bytes

Union variable: Define Union?

```
union [union tag] {
   member definition;
   member definition;
   ...
   member definition;
} [one or more union variables];
```

Union variable?

- Declare Union variables:

union <union tag> <union variable>;

- Size of Union?

The memory occupied by a union will be large enough to hold the largest member of the union.

- Accessing Union Members.

*To access any member of a union, we use the **member access operator (.)**.*

- User case?

Enum variable

- What is enum?
- Define enum.
- Declare enum variables.
- Size of enum?
- User case?

Enum variable: What is enum?

Enumeration is a user defined datatype in C language. It is used to assign names to the integral constants which makes a program easy to read and maintain. The keyword “enum” is used to declare an enumeration.

(wiki)

Enum variable

- Define enum:

The syntax of enum in C language:

enum enum_name{const1, const2, ... };

- Declare enum variables:

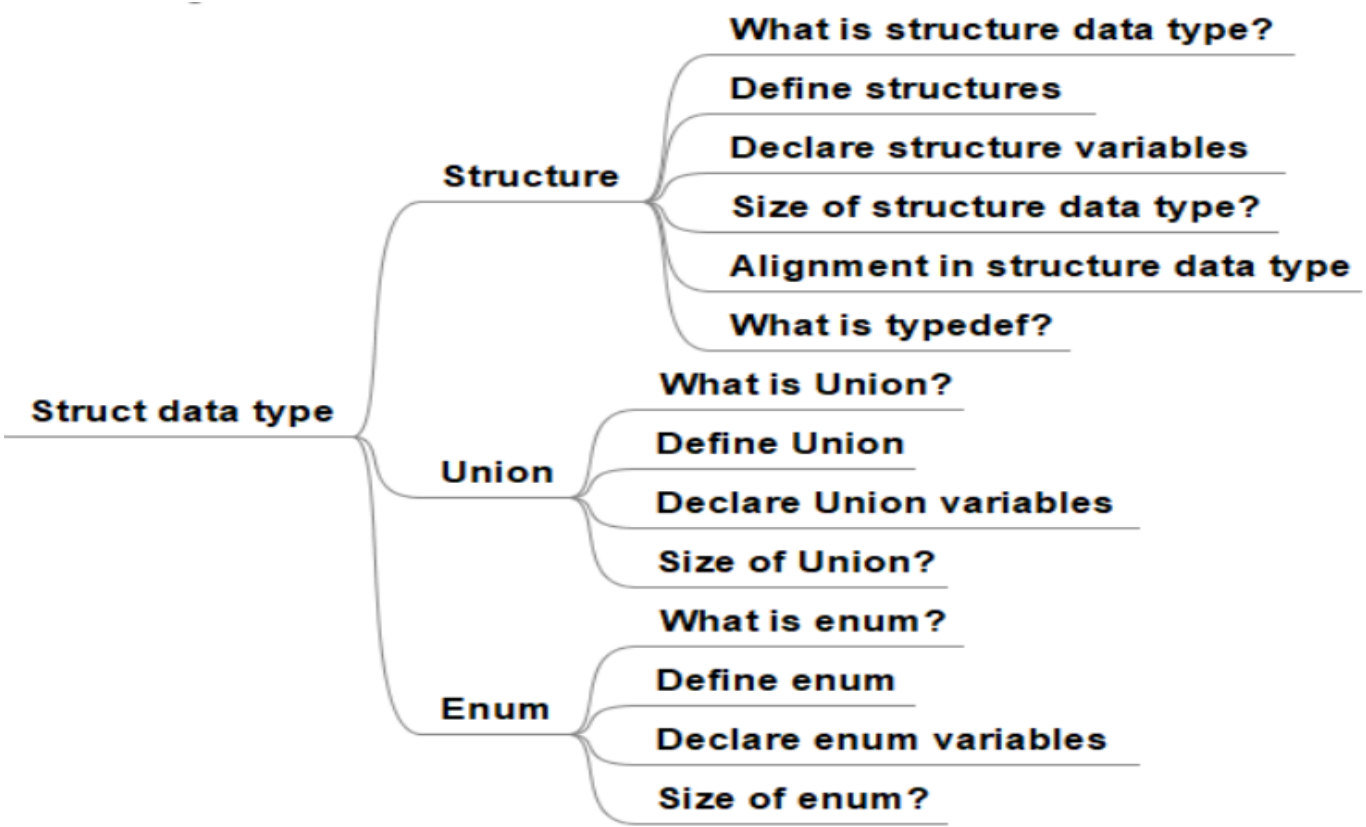
enum enum_name enum_variable;

- Size of enum?

In C all enums are most of the time integers of type int.

- User case?

Structure data type. Summary



Lesson Summary

- Basic Data Types
- Store Class
- Key words for variable
- Pointer variable
- Struct data type

Thank you