Ch 8 Structures, Unions, and Enumerations

An enum is a type with a set of named constants.

8.2 Structures

Variables of struct types can be initialized using {} notion.
Members are allocated in memory in declaration order.
The size of an object of a struct is not necessarily the sum of the sizes of its members.

You can minimize the wasted space by simply ordering members by size.
(largest member first)

The name of a type becomes available for use immediately after it has been encountered and not just after the complete declaration has been seen.

A struct is a class where the members are public by default.

std::memcpy() block-move machine instruction.

POD = "Plain Old Data" is an object that can be manipulated as just data without worrying about complications about class layouts or user-defined semantics.

Trivial type : a trivial default constructor
trivial move and copy operations

A union is a struct in which all members are allocated at the same address so that the union occupies only as much space as its largest member.

```
union Value {
        char* s;
        int i;
    };
```

Unions and Classes

Tagged union; discriminated union.

Enumerations

- Enum classes: enumerator names are local to the enum and values do not implicitly convert to other types
- Plain enums: enumerator names are in the same scope as the enum and their values implicitly convert to integers.

By default, enumerator values are assigned increasing from 0.

The plain "enum" enumerators are exported into the enum's scope.

The injection of names into an enclosing scope is namespace pollution.

The sizeof an enumeration is the sizeof its underlying type.

A plain enum can be unnamed, we use that when all we need is a set of integer constants, rather than a type to use for variables.