

سلسلة تعلم البرمجة بلغة C++ الحديثة

Learn Modern C++ Programming Course

إعداد المهندس أحمد الديب



#12: Unions

Unions

- 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 Status {
    unsigned int reg;
    struct {
        unsigned int TXE : 1;    // Transmit data register empty
        unsigned int TC : 1;    // Transmission complete
        unsigned int RXNE : 1;  // Read data register not empty
        unsigned int ERR : 3;   // Error
        unsigned int : 26;     // Reserved
    } bitfields;
};
```

Wasting Memory

```
enum Type { str, num };
struct Entry {
    char const* name;
    Type t;
    char const* s; // use s if t==str
    int i;         // use i if t==num
};

void print_entry(Entry& p) {
    std::cout << p.name << " = ";

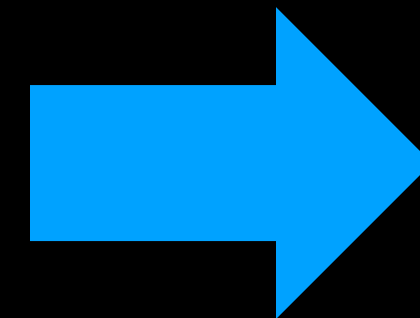
    if (p.t == str) {
        std::cout << p.s << "\n";
    } else {
        std::cout << p.i << "\n";
    }
}

int main() {
    Entry entry1{.name = "Car Manufacturer", .t = str, .s = "BMW"};
    Entry entry2{.name = "Person Height", .t = num, .i = 185};

    print_entry(entry1);
    print_entry(entry2);
}
```

Using Unions

```
enum Type { str, num };
struct Entry {
    char const* name;
    Type t;
    char const* s; // use s if t==str
    int i;         // use i if t==num
};
```



```
enum Type { str, num };
struct Entry {
    char const* name;
    Type t;
    union {
        char const* s;
        int i;
    } value;
};
```

Misusing Unions

```
union dirty_stuff {
    long int i;
    long int* p;
};

int main() {
    long int x{0x10};

    dirty_stuff bad{.p = &x};
    std::cout << "bad.i = " << std::hex << bad.i << std::endl;

    dirty_stuff dangerous{.i = bad.i};
    std::cout << "*dangerous.p = " << *dangerous.p << std::endl;

    // less bad than above is to use explicit type conversion operator
    auto pointer = reinterpret_cast<long int*>(bad.i);
    std::cout << "*pointer = " << *pointer << std::endl;
}
```

Thank you