Programming and Problem Solving
Union in C

R.M.U.A. Rathnayake

What is a Union?

- Union is very much similar to structure in c
- A collection of different data types grouped into a whole unit
- Each element in the group is called a member

Union vs Structure

- Structure helps to construct a complex data type which is more meaningful
- An array can be seen as a similar structure which holds multiple items
- How it differs from array ??

Differences between Union and Structure

C Structure	C Union
Structure allocates storage space for all its members separately	Union allocates one common storage space for all its members. Union finds that which of its member needs high storage space over other members and allocates that much space
Structure occupies higher memory space	Union occupies lower memory space over structure.
We can access all members of structure at a time.	We can access only one member of union at a time.

Differences between Union and Structure

C Structure	C Union

Structure example:
 	
struct student
{
int mark;
char name[6];
double average;
};
	Union example:
	
	union student
	{
	int mark;
	char name[6];
	double average;
	};

Differences between Union and Structure

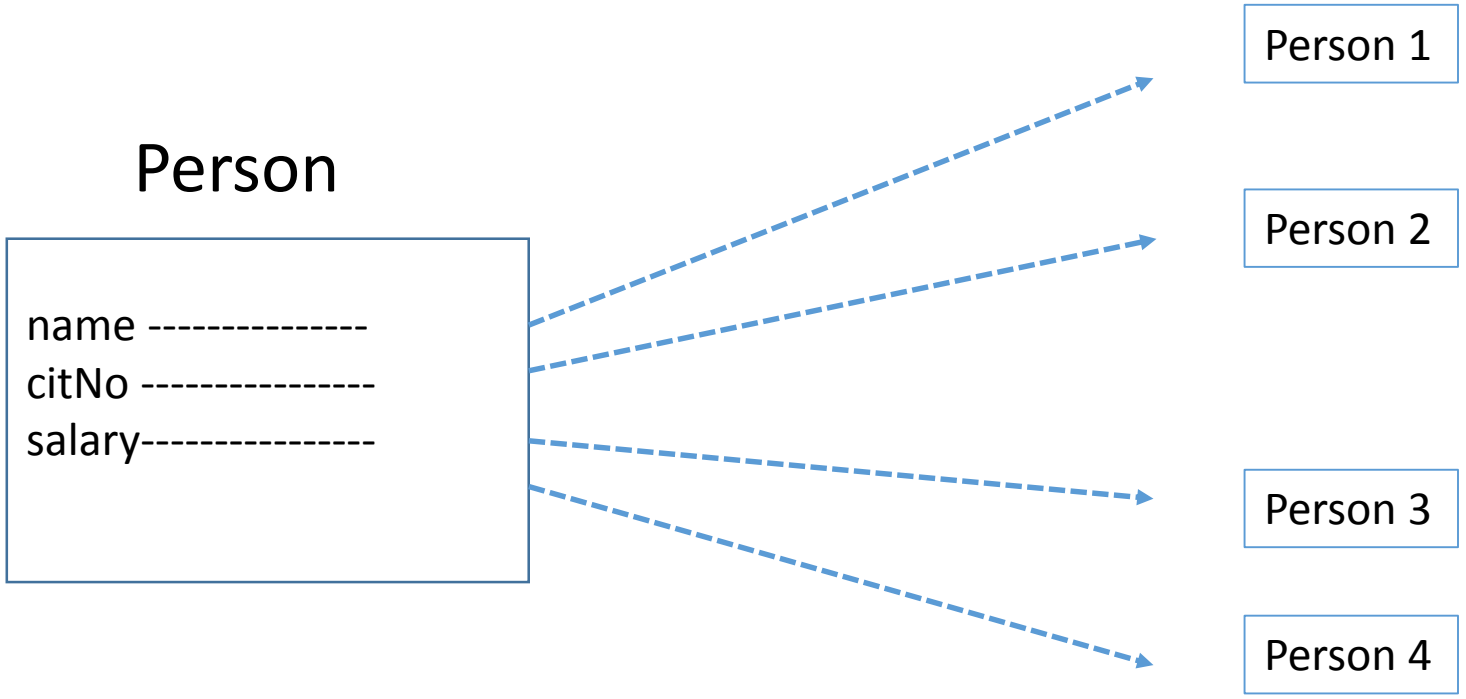
C Structure	C Union
For above structure, memory allocation will be like below. int mark – 2B char name[6] – 6B double average – 8B Total memory allocation = 2+6+8 = 16 Bytes	For above union, only 8 bytes of memory will be allocated since double data type will occupy maximum space of memory over other data types. Total memory allocation = 8 Bytes

Why union in C

- Union allow to store multiple data related to a particular concept
  - Eg: store information about a person: name, citizenship number, and salary
- You can create different members name, citNo and salary to store this information
- But if you need to store information of more than one person?
  - Need to create different variables for each information per person
  - Eg: name1, citNo1, salary1, name2, citNo2, salary2, etc.
  - What if the number of persons increase? (imagine 1000 for example)

Why Union in C

- A better approach would be to have a collection of all related information under a single name Person Union and use it for every person



Defining a Union type

- Union type definitions should appear near the top of a program file
- Should be outside of any function definition
- “union” keyword is used to define union
- Boundary of the union member definitions can be identified by “{“ and “}” symbols

Syntax to define a union in C

```
Union <unionname> {
        <member type> <member 1 name>;
        <member type> <member 2 name>;
        <member type> <member 3 name>;
        ...
};
```

Example

- Here is an example of defining a new type 'struct student' for storing student data

```
union student {
        char name[64];
        int  age;
        int  gradYear;
        float gpa;
};
```

```
union car{
        char name[50];
        int price;
        char model[20];
};
```

When you write the program

```c
#include <stdio.h>

union Car{
   char name[20];
   char make[20];
   char model[20];
   float price;
};


int main()
{
    return 0;
}
```

Union Variables

- Variables used to hold individual data
- When a union type is declared, no storage or memory is allocated
- To allocate memory of a given union type and work with it, we need to create variables

Creating union variables

union student student1;

student1 is a union type of **student**

student student2;

student2 is also a union **student** (we are just using the typedef alias name)

student csStd[10];

an array of **student** union: each bucket stores a studentT union

Example 1

```
union person {
        char name[50];
        int citNo;
        float salary;
};
int main() {
        union person person1, person2, p[20];
        return 0;
}
```

Union definition

Creating variables of union

Example 2

```
union person {
        char name[50];
        int citNo;
        float salary;
} person1, person2, p[20];
```

In both cases, two variables person1, person2, and an array variable p having 20 elements of type union “person” will be created

How to access members of a union

- There are two types of operators used for accessing members of a union.

1. . - member operator
2. -> - union pointer operator

- If you want to access the salary of person2. Here's how you can do it.

```
person2.salary
```