C Programming

Trainer : Smita Kadam

Email ID : smita@sunbeaminfo.com

Structure

- Keyword struct is used to declare user defined data type.

- Is a collection of similar/ dissimilar type of member fields and bit fields.
- **Syntax:**

```
struct <tagname>
{
        <data type> <identifier>;
        .....
        <signed/unsigned> <identifier> : <no.of bits>;
};
```

- Each member/field in structure receives memory separately and in sequence.

- Structure member/field can be accessed outside declaration via structure variable.

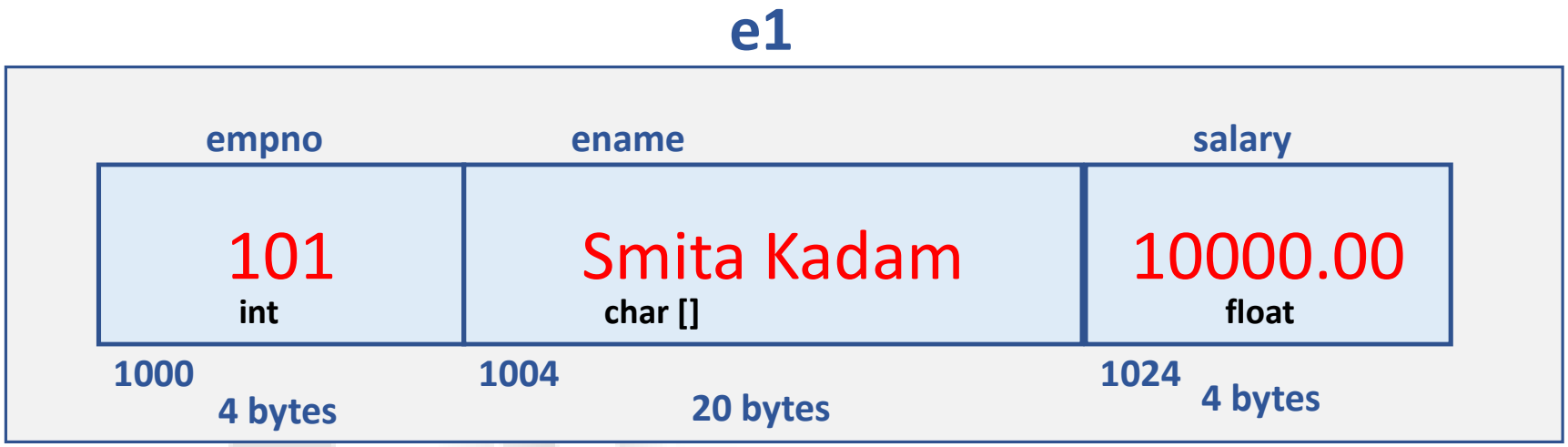
Structure

- **Declaration**

```
struct emp
{
        int empno;
        char ename[20];
        float salary;
}e1;     //declares e1 variable of struct emp type
```

- struct emp e2; //declares e2 variable of struct emp type.
- typedef struct emp EMPLOYEE; //helps to give alias to struct emp type.
- EMPLOYEE e3; // declares e3 variable of struct emp / EMPLOYEE type.

Memory Skeleton and How to Access structure members



28 bytes

```
e1.empno = 101
strcpy(e1.ename,"Smita Kadam");
e1.salary = 10000.00
```

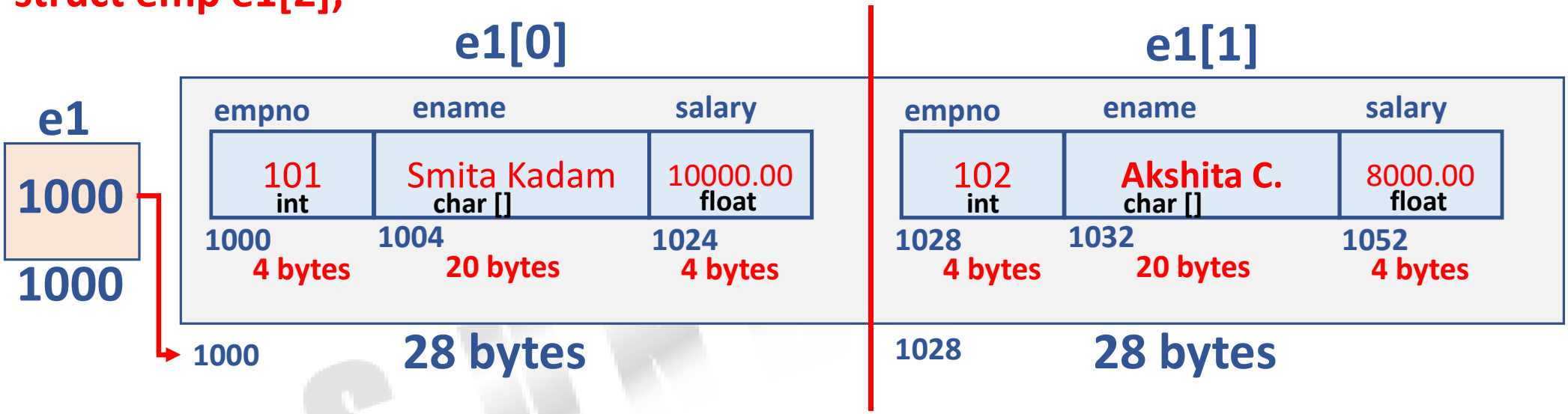
Nested Structure

- **Declaration**

```
struct emp
{
        int empno;
        char ename[20];
        float salary;
        struct date
        {
                int day,month,year;
        }dob;
}e1;
```

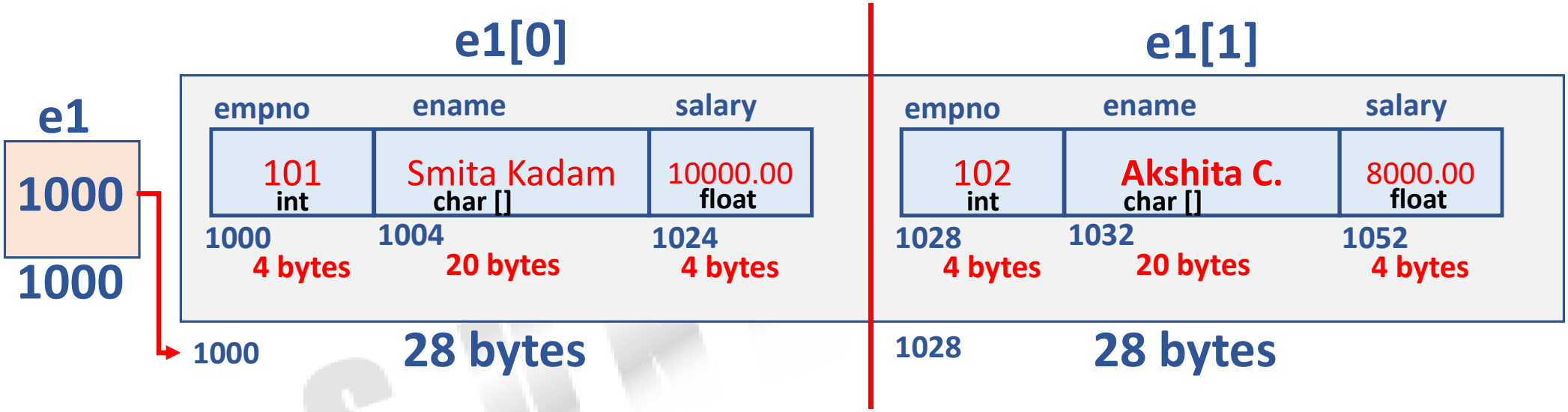
Array using structure

struct emp e1[2];




```
e1[0].empno = 101
strcpy(e1[0].ename,"Smita Kadam");
e1[0].salary = 10000.00
```

Array using pointer




```
struct emp *p = e1;
(p+0)->empno = 101
strcpy((p+0)->ename,"Smita Kadam");
(*(p+0)).salary = 10000.00
```

Bit Field

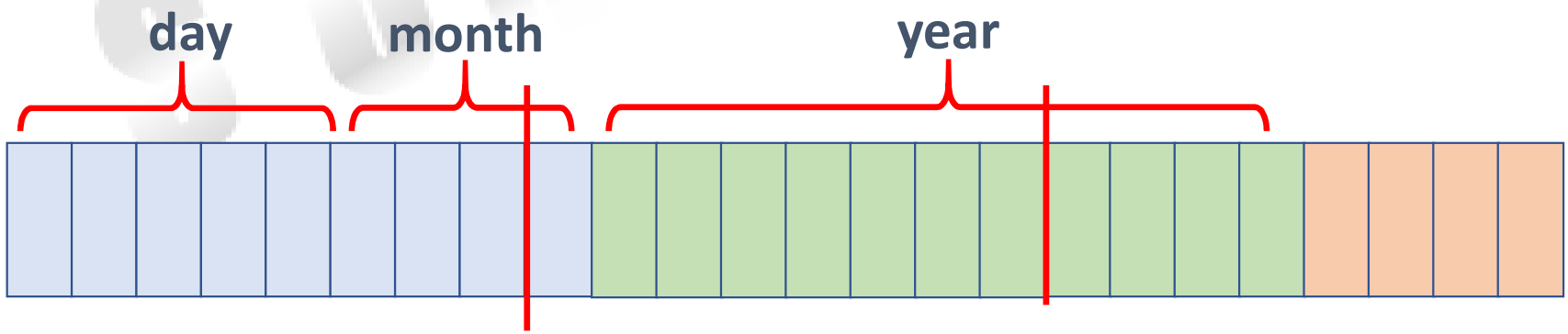
- **Syntax**

  <signed>/<unsigned> <identifier> : <no. of bits> ;

- Can be declared only within structure.
- Helps to identify bit / bits.
- Can not apply address operator on bit field.
- Should be initialised with value which can fit into specified bits else will result into unpredictable value.

Bit Field Declaration and Memory Allocation

```
#pragma pack(1)
struct date
{
        unsigned day : 5;
        unsigned month : 4;
        unsigned year : 11;
};
```

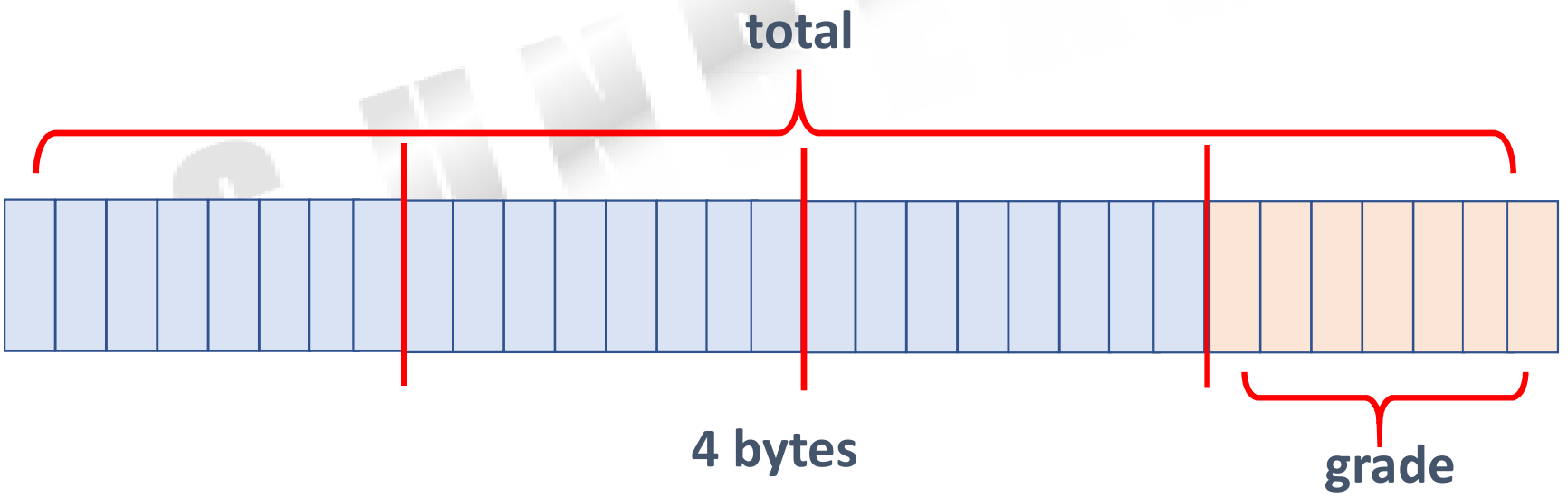


union

- **Points to note:**
  - Can be used to declare user data type.
  - Can be collection of similar/dissimilar type, bit field collection
  - Memory is shared amongst all members.
  - Memory is allocated as per the requirement of a member who need highest memory.
  - Latest assigned member can be accessed in right manner.

Union memory allocation

```
union result
{
        char grade;
        int total;
}r1;
```



Thank you!