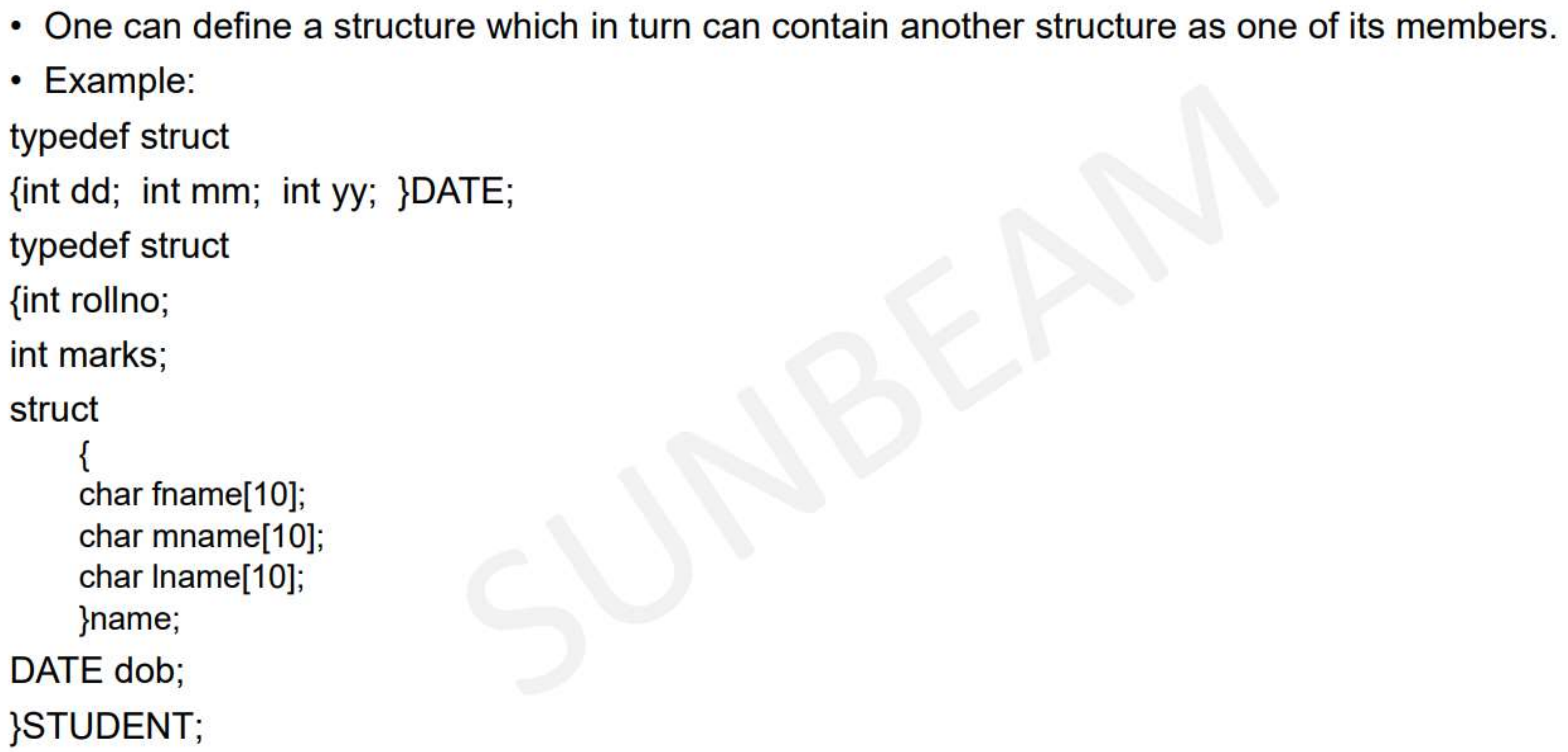
C programming

Trainer : Nisha Dingare

Email : nisha.dingare@sunbeaminfo.com

Nested Structure :



- One can define a structure which in turn can contain another structure as one of its members.
- Example:

```
typedef struct
{int dd;  int mm;  int yy;  }DATE;
typedef struct
{int rollno;
int marks;
struct
        {
        char fname[10];
        char mname[10];
        char lname[10];
        }name;
DATE dob;
}STUDENT;
```

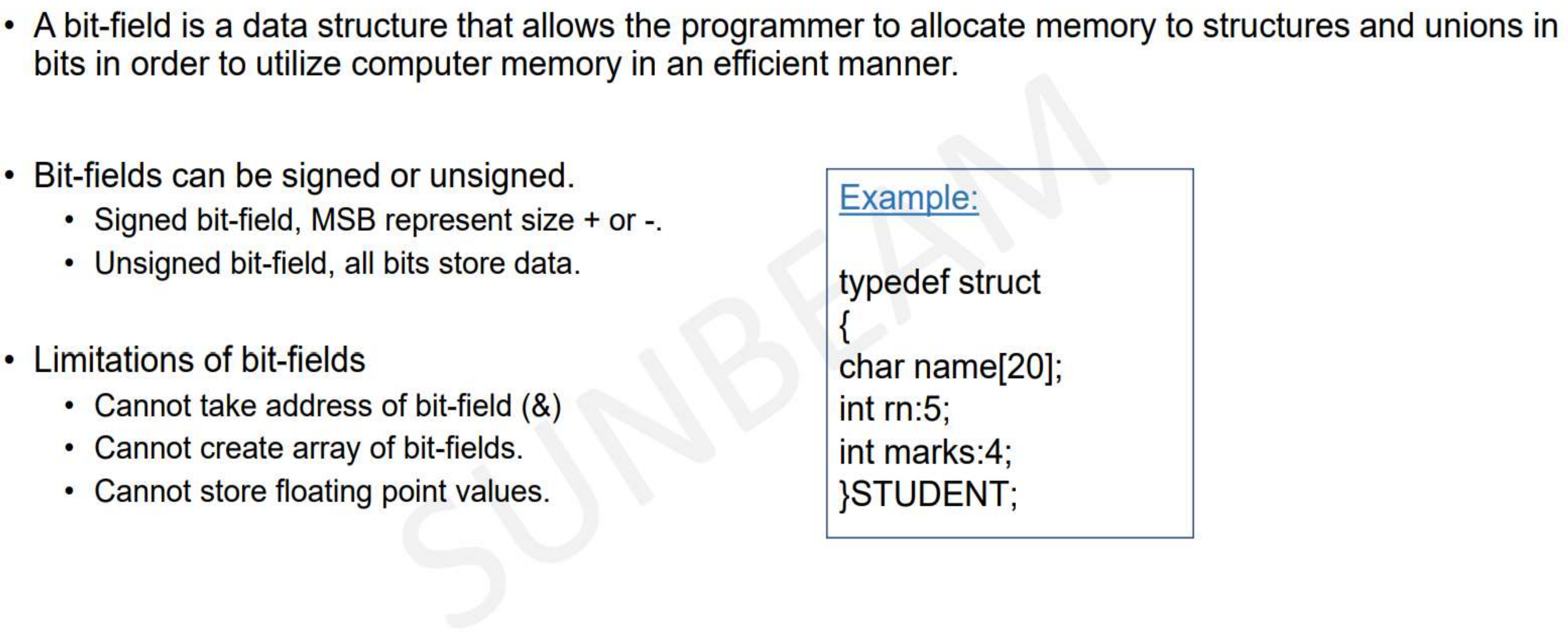
Structure Padding :

- For efficient access compiler may add hidden bytes into the struct called as "struct padding" or "slack bytes".
- On x86 architecture compiler add slack bytes to make struct size multiple of 4 bytes (word size).
- These slack bytes not meant to be accessed by the program.
- Programmer may choose to turn off this feature by using #pragma.
    - #pragma pack(1)

```
struct test {
    int a;
    char b;
};
printf("%u\n", sizeof(struct test));

#pragma pack(1)
struct test {
    int a;
    char b;
};
printf("%u\n", sizeof(struct test));
```

Bitfields :



- A bit-field is a data structure that allows the programmer to allocate memory to structures and unions in bits in order to utilize computer memory in an efficient manner.

- Bit-fields can be signed or unsigned.
  - Signed bit-field, MSB represent size + or -.
  - Unsigned bit-field, all bits store data.

- Limitations of bit-fields
  - Cannot take address of bit-field (&)
  - Cannot create array of bit-fields.
  - Cannot store floating point values.

Example:

```
typedef struct
{
char name[20];
int rn:5;
int marks:4;
}STUDENT;
```

Union :

- Union is user defined data-type.
- Like struct it is collection of similar or non-similar data elements.
- All members of union share same memory space i.e. modification of an member can affect others too.
- Size of union = Size of largest element
- When union is initialized at declaration, the first member is initialized.
- Application:
  - System programming: to simulate register sharing in the hardware.
  - Application programming: to use single member of union as per requirement.

```
union test
{
    int num;
    char arr[2];
}u = { 65 };


printf("%d, %c, %s\n", u.num, u.arr[0], u.arr);
```

Unions and structures :

- Unions and structures can be nested within each other.

```
struct
{
		short s[5];
		union
		{
		char x;
		float y;
		long z;
		short int z1;
		}u;
}t;
```

Thank You !!