Structure as Parameter

- If we are sending structure as a parameter to a structure it may be call by value or call by address

call by value

- Example

```
int area(struct rectangle r1)
{
    return r1.length*r1.breath;
}

int main()
{
    struct rectangle r = {10,5};
    printf("% d", area(r));
}
```

- A separate object will be created in call by value method and everything will be copied in the corresponding members this is one of the befit of structure as parameter

- If you are making changes to formal parameter it will not effect the actual parameter

call by reference

- Example

```
int area(struct rectangle  &r1)
{
    return r1.length*r1.breath;
}

int main()
{
    struct rectangle r = {10,5};
    printf("% d", area(r));
}
```

- The only change we need to do for call by reference is adding & in the parameter
- Here the new object is not created but the ***same object is called r1*** also
- Thus new changes in the values will effect the actual parameters

call by address

- Example

```
Void changeLength(struct rectangle *p , int l)

{
    p -> length = l;
}

int main()
{
    struct rectangle r = {10,5};
    ChangeLength(&r, 20);
}
```

- If you want some function to change the actual parameter then it must be done by call by address or call by reference

- It is possible to send array as a parameter in pass by value **only** if its inside the **structure** , if its just array passing then its not possible by pass by value