Base Class Ptr & Base Class reference

(Unit-4)

~~Normally~~, ptr → pointer
var → variable
ref → reference
bcp → base class ptr.
dco → der class obj

bcp/r → base class ptr/ref.

• Normally, ptr & var shud be of same types

eg
```
int a = 30;
int *p;
p = &a
```
p → int ptr
&a → int var

```
Char c = 'a';
int *p;
p = &c;
```
p → int ptr
&c → Char var.

✗ Error: Not allowed.

• Similar, rule for references.

eg
```
int a = 10;
int &ref = a;
```
ref → int ref
a → int var
✓

```
Char c = 'a';
int &ref = c;
```
ref → int ref
c → Char var
✗ Not allowed.

- However, when it comes to inh., a base ptr can point to a der class obj
- Note that base class ptr & der class obj r of different types, still bcp can point to a dco.
- But, opposite is not true. i.e a der class ptr can't pt. to a base class obj
- Same is applicable to base class ref. as well.

Eg 1 bcp pointing to base class obj.

```
base obj_b; //create b obj
base *bptr; // " " ptr.
bptr = &obj_b; // bptr pointing to
                  b obj.
```

Eg-2 der class ptr pointing to der class obj.

```
d2 obj_d2;
d2 *d2ptr;
d2ptr = &obj_d2; //d2 ptr point
                    to d2 obj.
```

Eg-3 bcp point to dco

```
d2 obj_d2;
base *bptr;
bptr = &obj_d2; // bptr point
                   to d2 obj
```

//Allowed

Eg-4

```
d2 * d2ptr;
base obj-b;
// d2ptr = &obj-b;
```

der ptr pointing to a base class obj

* This is not allowed. *

Similarly a base class ref can point to a der class obj.

eg-5

```
d2 obj-d2;
base &ref = obj-d2; // base ref to
                       a d2 obj.
```

Summary

base ptr/ref → Can point to its own class obj.

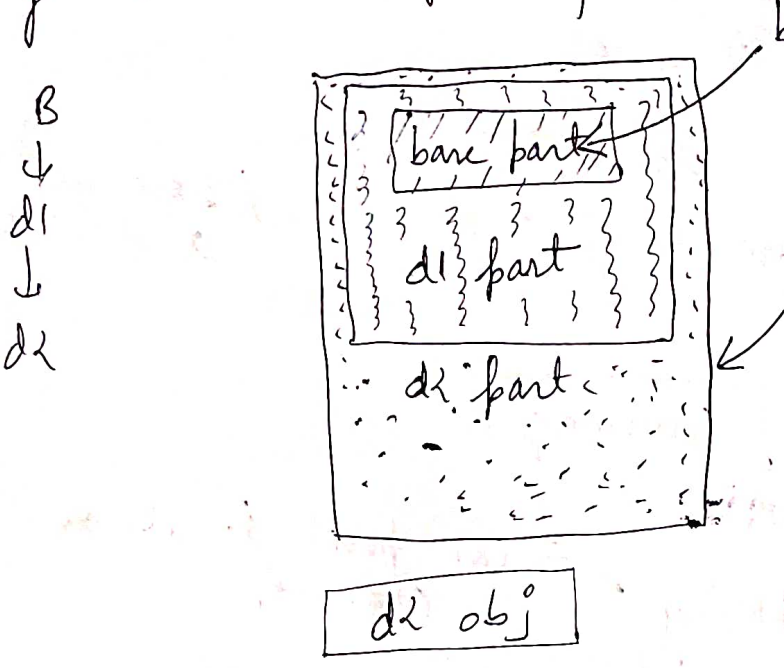
→ Can also pt. to its der class obj.

der class ptr/ref → Can pt. to its own class obj

→ CAN'T pt. to its base class obj.

Q) Why does a bcp/r can point to its der class obj but a der class ptr can't point to its base class obj. ?

A A ptr must know the size of block of m/m it is pointing to.



- In this eg:- obj of d2 is made of three parts
  1) base part : ~~der~~ inherited from base.
  2) d1 " : " " d1
  3) d2 " : Contains complete d2 obj.

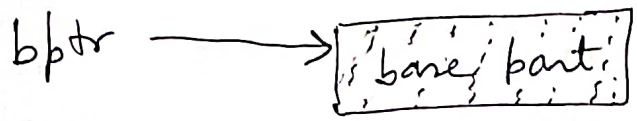
- When a bptr is pointing to this obj, then it is pointing to only the base ~~ptr~~ part. Because this is how ptr behaves. bptr knows only this much block of m/m.

- when a d2ptr is pointing to this obj then it is pointing to the complete d2 part. because a d2ptr knows the complete block of m/m.

- ∴ Both, d2ptr & bptr can point to a d2 obj.

Note d1 ptr can also point to d2 obj.

Now lets take another eg.



obj of base.

- Suppose there is an obj of base. It has only base part, it doesn't contain d1 & d2 parts.
- So only a base class ptr can point to a base class obj.
  A der class ptr can't pt. to a base class obj, bcoz base class obj do not have any der class parts.