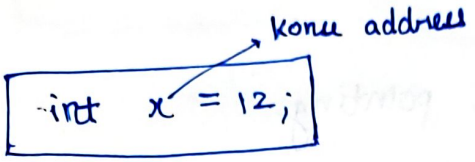
Fri, 3 March

Week6:
Lec1:

Pointers – Level1

① Pointers



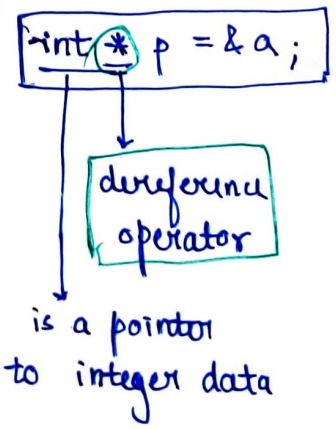
& → address of

```
int main(){
    int a = 5;
    cout<<a<<endl;
    cout<<&a<<endl;
    return 0;
}
```

Output:

```
5
0x7fff46810264
```

int a = 5, ✓



char * p = &ch;

↓

p is pointer to char data

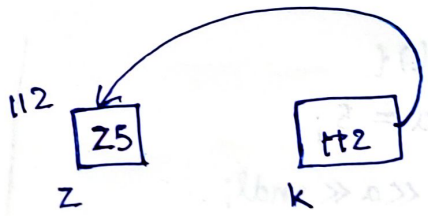
bool * p = &-;

↓

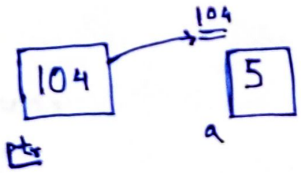
p is a pointer to bool data

```
int main(){

    int a = 5

    // create pointer

    int * ptr = &a;

    // access value ptr is pointing

    cout << *ptr << endl;
```

```
int z = 25;

int * k = &z;
```




```
int  a=5;

int* ptr = &a;
```



ptr ⟶ address

ptr ⟶ value

*ptr

↓

value at location stored in ptr

Creation

↳ int *p = &a;

access

↳ address → p

value → *p

```
int main(){
    int a=5;
    int* ptr=&a;
    cout<<"Address of a is: "<<&a<<endl;
    cout<<ptr<<endl;
    cout<<*ptr<<endl;
    cout<<&ptr<<endl;
}
```

Output

Address of a is: Location1

Location1

5

Prints address of ptr

NOTE:

*ptr → value stored at ptr location

&ptr → address of ptr

&a → address of a

a → value of a

ptr → value of ptr

Q.

```
int a=5;                    int *p = &a; ⟶ size ⟶ 4
char ch='b';                char *c = &ch; ⟶ size .
double d=1.03;              double * dptr = &d ⟶ size
```

O/P

```
8
8
8
```

- Pointer ~~dat~~ size '8' aaega [depends on computer architecture]

Why 8?

H.W→ What is meaning of 64 bit system?

✗ Bad practice
```
int* ptr;
cout<<*ptr<<endl;
```
SEGMENTATION ERROR

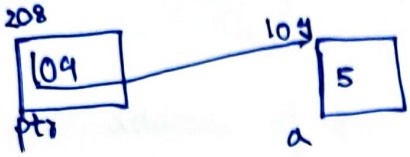
✓
```
int* ptr = 0 //NULL POINTER
cout<<*ptr<<endl;
```

O/P

Segmentation fault

↓

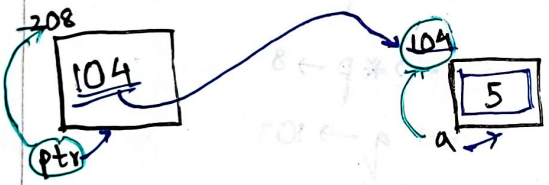
अपनी Memory access kro humari mat chedo



a = a + 1;

p = p + 1;

104 + 1 = 108



ptr

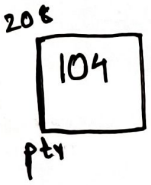
a →

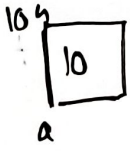
*ptr

& ptr

&a

Ques :-





i) a → 10

ii) &a → 104

iii) ptr → 104

iv) *ptr → 10

v) &ptr → 208

vi) *p * 2 → 20

vii) (*ptr)++ → 11

viii) ++(*ptr) — 12

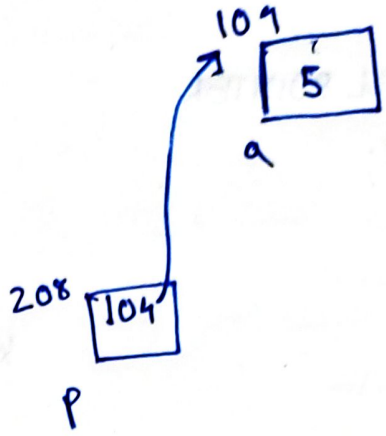
(ix) a = a + 1 — 13

(x) *p = *p + 2 — 15

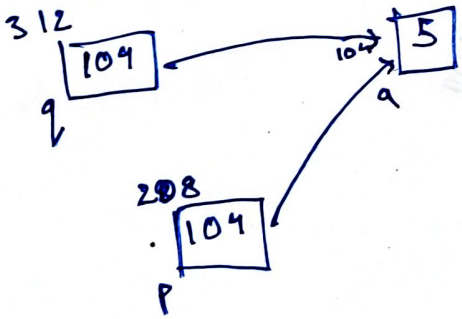
(xi) *p = *p x 2 — 30

(xii) *p = *p / 2 — 15

```
int a = 5;
int *p = &a
int *q = p;
```




```
int a = 5;
int *p = &a;
int *q = p;
```



→ a – 5

→ &a – 104

→ p – 104

→ &p = 208

→ *p → 8

→ q → 104

→ &q – 312

→ *q – 5

→ *p/2 – 5/2

→ *q/2 – 5/2

```
int a=10;
int* p = &a;
int* q = p;
int* r = q;
```

a — 10

&a — address of a

p — " " "

&p — address of p

*p — 10

q — address of a

&q — address of q

*q — 10

r — address of a

&r — address of r

*r — 10

(*p+*q+*r) — 30

(*p)*2+(*r)*3 — 50

(*p)/2-(*q)/2 — 0