

Con trỏ

Khai báo biến con trỏ

- Kiểu dữ liệu * tên biến con trỏ;
- Khởi tạo một vùng sẽ giữ địa chỉ một vùng nhớ
- Lưu ý :
- Biến Con trỏ cần phải có khởi tạo (cần phải giữ một địa chỉ)

Gán giá trị cho con trỏ

- Gán giá trị cho con trỏ là gán địa chỉ của một vùng nhớ cho biến con trỏ
- Có các cách
- Khởi tạo biến con trỏ -New
new(Tenbiencontro)
- Gán địa chỉ
BiếnContrỏ=địa chỉ
- Gán bằng địa chỉ của một biến khác có cùng kiểu
Biencontrỏ=&tênbiênkhác

*Chú ý : địa chỉ của 1 biến được truy xuất
&tênbiến*

Tham chiếu con trỏ

- *tenbiến con trỏ

Sẽ lấy giá trị tại địa chỉ mà con trỏ đang giữ

Ví dụ

```
Int a =5;
```

```
Int *p=&a;
```

```
*p → 5;
```

Ví dụ 1

- `char ch;`
- `float fl;`
- `printf("&ch la %u va &ch+1 la %u",
&ch, &ch + 1);`
- `printf("&fl la %u va &fl+1 la %u", &fl,
&fl + 1);`

Ví dụ 2

- `Int X=5;`
- `Int *p;`
- `P=&x;`
- `Cout<<*p,x; →?`
- `*p=10;`
- `Cout<<*p,x; →?`

Ví dụ 3

- `Int X=10;`
- `Int *p;`
- `Cout<<*p,x; →?`
- `*p=10;`
- `Cout<<*p,x; →?`

Ví dụ 4

- `Int X=5;`
- `Int *p;`
- `P=&x`
- `Cout<<&x; →?`
- `x=10;`
- `Cout<<*p,x; →?`

Ví dụ 5

- `Int X=5;`
- `Int *p,*q;`
- `New(p);`
- `Q=&x;`
- `*p=x;`
- `Cout<<x<<*p<<*q; →?`
- `*p=10;`
- `Cout<<x<<*p<<*q; →?`
- `*q=*p;`
- `Cout<<x<<*p<<*q; →?`

Bộ nhớ

	3	5	7	2	88	1	1	12
	0x01f	0x02	0x02	0x02	0x02	0x02	0x02	0x02
		0	1	2	3	4	5	6
int x;	12	12	90	11	1	33	1	45
	0x02	0x02	0x02	0x02	0x02	0x02	0x02	0x02
X=9;	7	8	9	a	b	c	d	e
	12	212	12	12	12	11	1212	1
	0x02f	0x03	0x03	0x03	0x03	0x03	0x03	0x03
&x → 0x03a		0	1	a	3	4	5	6
	12	0	77				56	67
	0x03	0x03	0x03	0x03	0x03	0x03	0x03	0x03
	7	8	9	a	b	c	d	e
	0x03f	0x04	0x04	0x04	0x04	0x04	0x04	0x04
		0	1	2	3	4	5	6

Bộ nhớ cho class

```
data {
  Int info;
  Data next;
};
```

```
data x;
X =new data();
x.info=9;
x.next= new data();
```

```
x.next.next->
NULL;
```

3	5	7	2	88	1	1	12
0x01f	0x02 0	0x02 1	0x02 2	0x02 3	0x02 4	0x02 5	0x02 6
12	12	90	11	1	33	1	45
0x02 7	0x02 8	0x02 9	0x02 a	0x02 b	0x02 c	0x02 d	0x02 e
12	212	12	12	12	11	1212	1
0x02f	0x03 0	0x03 1	0x03 a	0x03 3	0x03 4	0x03 5	0x03 6
12	0	77					67
0x03 7	0x03 8	0x03 9	0x03 a	0x03 b	0x03 c	0x03 d	0x03 e

x	x.next							
0x03 a	0x045							

Bộ nhớ cho class

```
data {
  Int info;
  Data next;
};
```

```
data x;
X =new data();
x.info=9;
x.next= new data();
```

```
Data p;
P=new data();
```

```
Data q;
q=x.next
;
```

```
Console.WriteLine(x.in
fo,x.next.info,q.info);
```

3	5	7	2	88	1	1	12
0x01f	0x02 0	0x02 1	0x02 2	0x02 3	0x02 4	0x02 5	0x02 6
12	12	90	11	1	33	1	45
0x02 7	0x02 8	0x02 9	0x02 a	0x02 b	0x02 c	0x02 d	0x02 e
12	212	12	12	12	11	1212	1
0x02f	0x03 0	0x03 1	0x03 a	0x03 3	0x03 4	0x03 5	0x03 6
12		77					67
0x03 7	0x03 8	0x03 9	0x03 a	0x03 b	0x03 c	0x03 d	0x03 e

x	x.next	p	q					
0x03 a	0x045	0x040	0x045					

- Data x ;
- Data p ;
- $P=x$; \rightarrow ????????

- Data x;
- X=new data();
- Data p;
- P=x; → ????????

- Data x;
- X=new data();
- Data p;
- P.info=x.info; → ????????

- Data x;
- X=new data();
- Data p;
- P=x;
- P.info =5;
- X.info =7;

- P.info → ????????