

Chương 11

ĐA XẠ – ĐA HÌNH – POLYMORPHISM

- Nguyễn Hữu Lợi
- Đoàn Chánh Thống
- ThS. Nguyễn Thành Hiệp
- ThS. Trương Quốc Dũng
- ThS. Võ Duy Nguyên
- ThS. Nguyễn Văn Toàn
- TS. Nguyễn Duy Khánh
- TS. Nguyễn Tấn Trần Minh Khang

Qui tắc 01

Qui tắc 1: Con trỏ đối tượng khi gọi thực hiện một phương thức, nếu phương thức đó là phương thức không ảo thì nó sẽ gọi thực hiện phương thức của lớp đối tượng mà con trỏ thuộc về.

Qui tắc 02

Qui tắc 2: Con trỏ đối tượng khi gọi thực hiện một phương thức, nếu phương thức đó là phương thức ảo thì chương trình sẽ xem xét xem con trỏ đối tượng đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng nào và chương trình sẽ gọi thực hiện phương thức của lớp đối tượng đó.

TRƯỜNG HỢP 04

Trường hợp 04

- Trường hợp 4: **XXXX** là virtual,
 YYYY là virtual.
- Chạy từng bước trường hợp 04.

Trường hợp 04

```
11.#include <iostream>
12.class A
13.{
14.    public:
15.        XXXX void Sketchy()
16.        {
17.            cout<<"\n A's Sketchy()";
18.            Sketchy(-1);
19.        }
20.        YYYY void Sketchy(int num)
21.        {
22.            cout<<"\n A's Sketchy("<<num<<")";
23.        }
24.};
```

Trường hợp 03

— Trường hợp 4: **XXXX** là virtual,
YYYY là virtual.

```
11.#include <iostream>
12.class A
13.{
14.    public:
15.        XXXX void Sketchy()
16.        {
17.            cout<<"\n A's Sketchy()";
18.            Sketchy(-1);
19.        }
20.        YYYY void Sketchy(int num)
21.        {
22.            cout<<"\n A's Sketchy("<<num<<")";
23.        }
24.};
```

Trường hợp 04

```
11.#include <iostream>
12.class A
13.{
14.    public:
15.        virtual void Sketchy()
16.        {
17.            cout<<"\n A's Sketchy()";
18.            Sketchy(-1);
19.        }
20.        virtual void Sketchy(int num)
21.        {
22.            cout<<"\n A's Sketchy("<<num<<")";
23.        }
24.};
```


Trường hợp 04

```
11.#include <iostream>
12.class A
13.{
14.    public:
15.        virtual void Sketchy()
16.        {
17.            cout<<"\n A's Sketchy()";
18.            Sketchy(-1);
19.        }
20.        virtual void Sketchy(int num)
21.        {
22.            cout<<"\n A's Sketchy("<<num<<")";
23.        }
24.};
```

Khai báo lớp đối tượng
A lớn

Trường hợp 04

```
11.#include <iostream>
12.class A
13.{
14.    public:
15.        virtual void Sketchy()
16.        {
17.            cout<<"\n A's Sketchy()";
18.            Sketchy(-1);
19.        }
20.        virtual void Sketchy(int num)
21.        {
22.            cout<<"\n A's Sketchy("<<num<<")";
23.        }
24.};
```

Một phương thức được khai báo bắt đầu bằng từ khóa **virtual** được gọi là phương thức ảo.

Trường hợp 04

```
11.#include <iostream>
12.class A
13.{
14.    public:
15.        virtual void Sketchy()
16.        {
17.            cout<<"\n A's Sketchy()";
18.            Sketchy(-1);
19.        }
20.        virtual void Sketchy(int num)
21.        {
22.            cout<<"\n A's Sketchy("<<num<<")";
23.        }
24.};
```

Phương thức Sketchy()
của lớp đối tượng A là
phương thức ảo.

Trường hợp 04

```
11.#include <iostream>
12.class A
13.{
14.    public:
15.        virtual void Sketchy()
16.        {
17.            cout<<"\n A's Sketchy()";
18.            Sketchy(-1);
19.        }
20.        virtual void Sketchy(int num)
21.        {
22.            cout<<"\n A's Sketchy("<<num<<")";
23.        }
24.};
```

Phương thức Sketchy(int)
của lớp đối tượng A là
phương thức ảo.

Trường hợp 04

```
25.class B:public A
26.{
27.    public:
11.        void Sketchy()
25.        {
26.            cout<<"\n B's Sketchy()";
27.            Sketchy(-2);
28.        }
29.        void Sketchy(int num)
30.        {
31.            cout<<"\n B's Sketchy("<<num<<")";
32.        }
33.};
```

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
11.         void Sketchy()
25.         {
26.             cout<<"\n B's Sketchy()";
27.             Sketchy(-2);
28.         }
29.         void Sketchy(int num)
30.         {
31.             cout<<"\n B's Sketchy("<<num<<")";
32.         }
33. };
```

Dòng số 25 được đọc là:
lớp đối tượng B kế thừa
từ lớp đối tượng A với từ
khóa dẫn xuất public.

Trường hợp 04

```
25.class B:public A
26.{
27.    public:
28.        void Sketchy()
29.        {
30.            cout<<"\n B's Sketchy()";
31.            Sketchy(-2);
32.        }
33.        void Sketchy(int num)
34.        {
35.            cout<<"\n B's Sketchy("<<num<<")";
36.        }
37.};
```

Phương thức Sketchy()
của lớp đối tượng B là
phương thức ảo.

Trường hợp 04

```
25.class B:public A
26.{
27.    public:
28.        void Sketchy()
29.        {
30.            cout<<"\n B's Sketchy()";
31.            Sketchy(-2);
32.        }
33.        void Sketchy(int num)
34.        {
35.            cout<<"\n B's Sketchy("<<num<<")";
36.        }
37.};
```

Phương thức **Sketchy(int)**
của lớp đối tượng **B** là
phương thức ảo.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Trường hợp 04

```
38. class C:public B
39. {
40.     public:
41.         void Sketchy(int num)
42.         {
43.             cout<<"\n C's Sketchy("<<num<<")";
44.         }
45. };
46. void Curious(A* wacky)
47. {
48.     wacky->Sketchy();
49.     ((C*)wacky)->Sketchy(123);
50. }
```

Dòng số 38 được đọc là:
lớp đối tượng **C** kế thừa
từ lớp đối tượng **B** với từ
khóa dẫn xuất **public**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Phương thức Sketchy(int)
của lớp đối tượng C là
phương thức ảo.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Dòng 46 định nghĩa hàm Curious.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Hàm **Curious** có một tham số đầu vào có tên là **wacky** và **wacky** là con trỏ đối tượng thuộc lớp **A** lớn.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Dòng 48 đọc là: con trỏ đối tượng **wacky** gọi thực hiện phương thức **Sketchy** không có tham số.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Dòng 49 đọc là: con trỏ đối tượng **wacky** được ép kiểu thành con trỏ đối tượng thuộc lớp **C**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Dòng 49 đọc là: con trỏ đối tượng **wacky** được ép kiểu thành con trỏ đối tượng thuộc lớp **C**.

Sau đó con trỏ đối tượng **wacky** gọi thực hiện phương thức **Sketchy** có tham số với đối số là **123**.

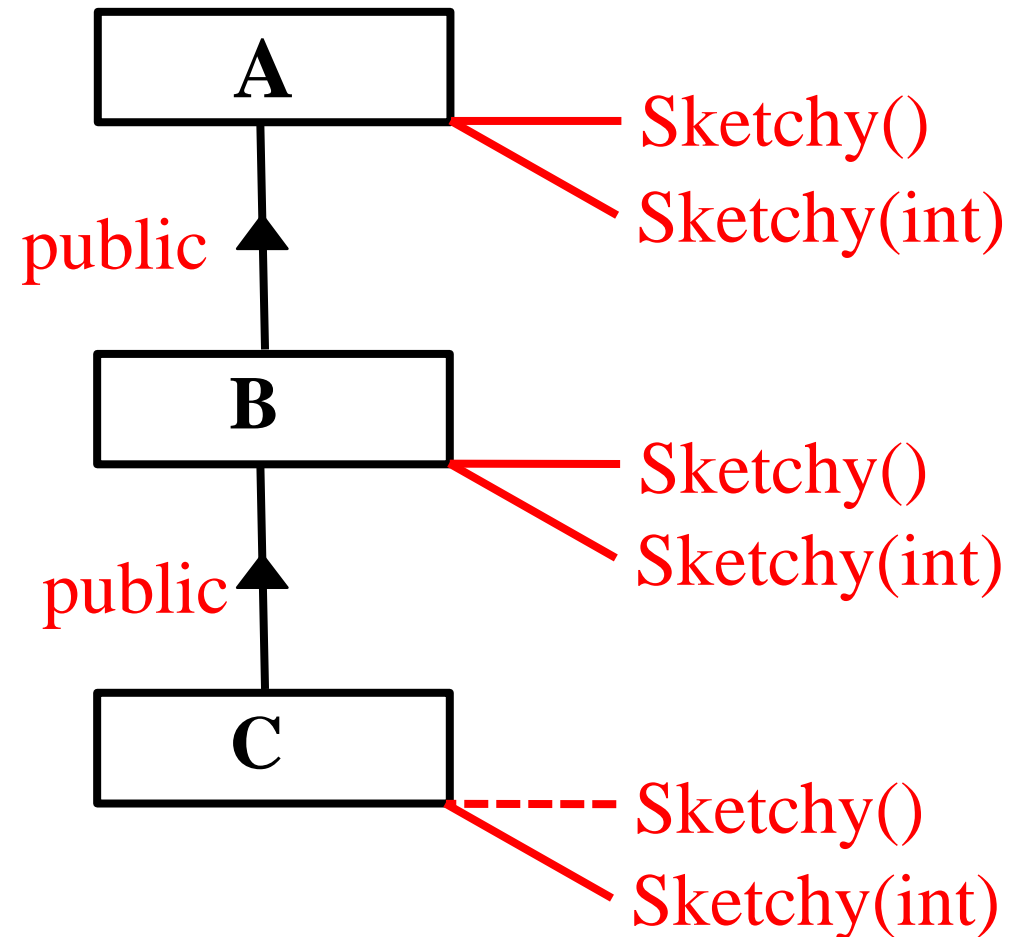
Trường hợp 04

```

51. void main()
52. {
53.     A* inky = new B;
54.     inky->Sketchy();
55.     inky->Sketchy(23);
56.     Curious(inky);

57.     B* pinky = new C;
58.     pinky->Sketchy();
59.     pinky->Sketchy(46);
60.     Curious(pinky);
61. }

```



Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 53 đọc là: **inky** là con trỏ đối tượng thuộc lớp đối tượng **A**.
- **new B** là tạo ra đối tượng thuộc lớp đối tượng **B**.
- Địa chỉ của đối tượng vừa được tạo ra được gán cho con trỏ đối tượng **inky**.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

— Dòng 54 đọc là con trỏ đối tượng **inky** gọi thực hiện phương thức **Sketchy()** không có tham số.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Đây là phương thức ảo.
- Do đó, chương trình xem xét xem con trỏ đối tượng **inky** đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng nào và nó sẽ gọi thực hiện phương thức của lớp đối tượng đó.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Con trỏ đối tượng **inky** đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng **B** nên phương thức **Sketchy()** của lớp đối tượng **B** được gọi thực hiện.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Câu lệnh đầu tiên trong thân phương thức Sketchy() của lớp đối tượng B là câu lệnh xuất liệu.

Dòng xuất liệu là:
B's Sketchy()

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Trong thân phương thức Sketchy() của lớp B có lời gọi thực hiện phương thức Sketchy(int) với đối số -2.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Đây là phương thức ảo.

Do đó, chương trình sẽ xem xét xem ai đang gọi thực hiện phương thức Sketchy() của lớp đối tượng B.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Câu trả lời là: con trỏ đối tượng inky đang gọi thực hiện phương thức Sketchy() của lớp đối tượng B.
- Con trỏ đối tượng inky đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng B.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Do đó, phương thức Sketchy(int) của lớp đối tượng B được gọi thực hiện.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Trong thân phương thức `Sketchy(int)` của lớp đối tượng `B` có một dòng xuất liệu.

Chương trình xuất thêm một dòng xuất liệu là:
`B's Sketchy(-2)`

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 54 được thực hiện xong.
- Chương trình xuất hai dòng xuất liệu là:
 - + B's Sketchy()
 - + B's Sketchy(-2)

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 55 đọc là con trỏ đối tượng **inky** gọi thực hiện phương thức **Sketchy(int)** có tham số với đối số là **23**.
- Đây là phương thức ảo.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

— Do đó, chương trình xem xét xem con trỏ đối tượng **inky** đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng nào và nó sẽ gọi thực hiện phương thức của lớp đối tượng đó.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Con trỏ đối tượng **inky** đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng **B**.
- Do đó, phương thức **Sketchy(int)** của lớp đối tượng **B** được gọi thực hiện.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Trong thân phương thức Sketchy(int) của lớp đối tượng B có một dòng xuất liệu.

Chương trình xuất thêm một dòng xuất liệu là:

B's Sketchy(23)

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 55 được thực hiện xong.
- Chương trình xuất thêm một dòng xuất liệu là:
+ B's Sketchy(23)

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 56 đọc là hàm **Curious** được gọi thực hiện với đối số là **inky**.
- **inky** là con trỏ đối tượng thuộc lớp đối tượng **A** và đang giữ địa chỉ của đối tượng về lớp đối tượng **B**

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Vào bên trong thân hàm
Curious đại diện của
inky là wacky.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Dòng 48 đọc là con trỏ đối tượng **wacky** gọi thực hiện phương thức **Sketchy()**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Đây là phương thức ảo.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Do đó, chương trình
xem xét xem wacky là
ai.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

wacky là tham số đầu
vào của hàm Curious.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Hàm **Curious** được gọi thực hiện với đối số là **inky**.
- **inky** là con trỏ đối tượng thuộc lớp đối tượng **A** và đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng **B**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Do đó, phương thức Sketchy() của lớp đối tượng B được gọi thực hiện.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Câu lệnh đầu tiên trong thân phương thức Sketchy() của lớp đối tượng B là câu lệnh xuất liệu.

Dòng xuất liệu là:
B's Sketchy()

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Trong thân phương thức Sketchy() của lớp B có lời gọi thực hiện phương thức Sketchy(int) với đối số -2.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Đây là phương thức ảo.

Do đó, chương trình sẽ xem xét xem ai đang gọi thực hiện phương thức Sketchy() của lớp đối tượng B.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu trả lời là: con trỏ đối tượng **wacky** đang gọi thực hiện phương thức **Sketchy()** của lớp đối tượng **B**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu hỏi kế tiếp là wacky là ai?

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu trả lời là: **wacky** là tham số đầu vào của hàm **Curious**.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Mà hàm **Curious** được gọi thực hiện với đối số là **inky**.
- **inky** là con trỏ đối tượng thuộc lớp đối tượng **A** và đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng **B**.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Do đó, chương trình sẽ gọi thực hiện phương thức `Sketchy(int)` của lớp đối tượng `B` với đối số là `-2`.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Trong thân phương thức Sketchy(int) của lớp đối tượng B có một dòng xuất liệu.

Chương trình xuất thêm một dòng xuất liệu là:

B's Sketchy(-2)

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu lệnh dòng 48 thực hiện xong.
Chương trình xuất hai dòng xuất
liệu là:

B's Sketchy()

B's Sketchy(-2)

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu lệnh dòng 49 đọc là **wacky** là con trỏ đối tượng thuộc lớp **A**, được ép kiểu thành con trỏ đối tượng thuộc lớp đối tượng **C**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu lệnh dòng 49 đọc là **wacky** là con trỏ đối tượng thuộc lớp **A**, được ép kiểu thành con trỏ đối tượng thuộc lớp đối tượng **C**.

Sau đó, gọi thực hiện phương thức **Sketchy(int)** với đối số **123**

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Đây là phương thức ảo.

Cơ chế đa xạ được ưu tiên hơn cơ chế ép kiểu.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Do đó, chương trình sẽ
xem xét xem **wacky** là ai?

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu trả lời là: **wacky** là tham số đầu vào của hàm **Curious**.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Mà hàm **Curious** được gọi thực hiện với đối số là **inky**.
- **inky** là con trỏ đối tượng thuộc lớp đối tượng **A** và đang giữ địa chỉ của đối tượng về lớp đối tượng **B**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Do đó, phương thức Sketchy(int) của lớp đối tượng B được thực hiện.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Trong thân phương thức `Sketchy(int)` của lớp đối tượng B có một dòng xuất liệu.

Chương trình xuất thêm một dòng xuất liệu là:
B's Sketchy(123)

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 56 được thực hiện xong.
- Chương trình xuất thêm ba dòng xuất liệu là:
 - + B's Sketchy()
 - + B's Sketchy(-2)
 - + B's Sketchy(123)

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 57 đọc là: **pinky** là con trỏ đối tượng thuộc lớp đối tượng **B**.
- **new C** là tạo ra đối tượng thuộc lớp đối tượng **C**.
- Địa chỉ của đối tượng vừa được tạo ra được gán cho con trỏ đối tượng **pinky**.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

— Dòng 58 đọc là con trỏ đối tượng **pinky** gọi thực hiện phương thức **Sketchy()** không có tham số.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Đây là phương thức ảo.
- Do đó, chương trình xem xét xem con trỏ đối tượng **pinky** đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng nào và nó sẽ gọi thực hiện phương thức của lớp đối tượng đó.

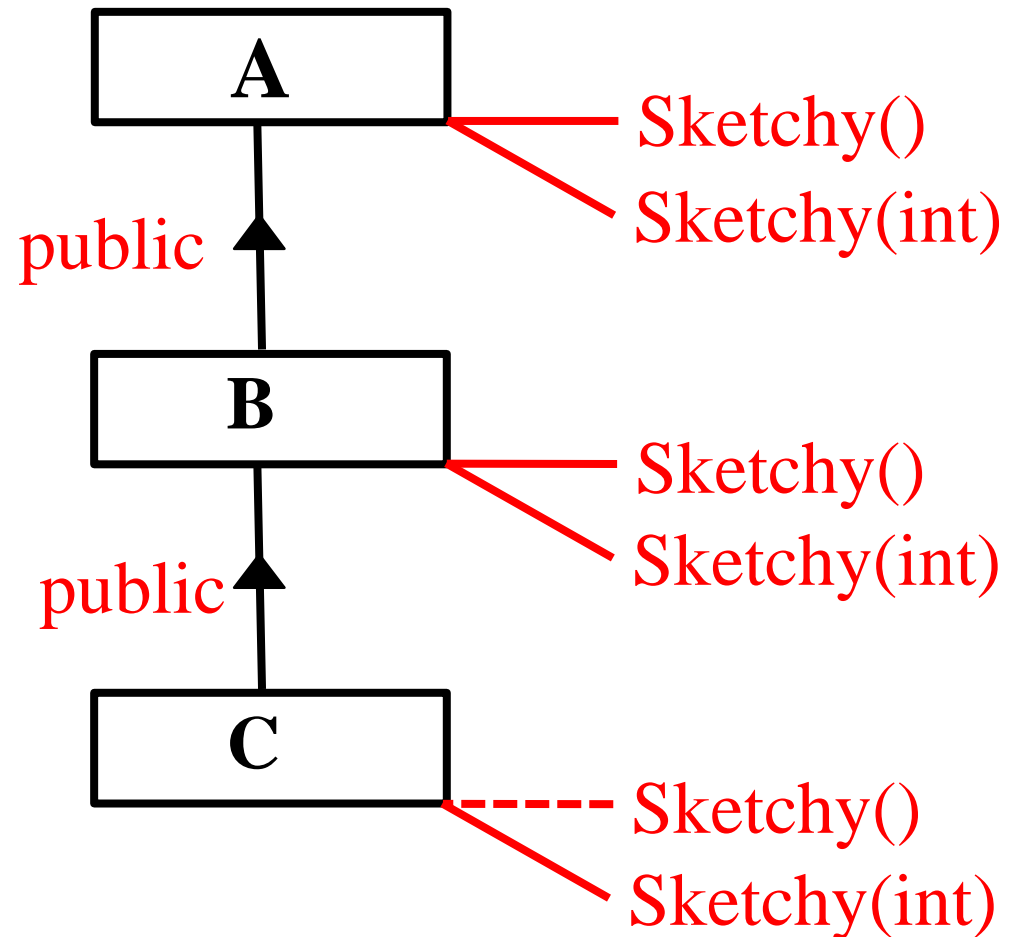
Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Con trỏ đối tượng **pinky** đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng **C** nên phương thức **Sketchy()** của lớp đối tượng **C** được gọi thực hiện.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```



Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

— Mà phương thức **Sketchy()** của lớp đối tượng **C** kế thừa từ lớp đối tượng **B**, nên phương thức **Sketchy()** của lớp đối tượng **B** được gọi thực hiện

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Câu lệnh đầu tiên trong thân phương thức `Sketchy()` của lớp đối tượng `B` là câu lệnh xuất liệu.

Dòng xuất liệu là:
`B's Sketchy()`

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Trong thân phương thức Sketchy() của lớp B có lời gọi thực hiện phương thức Sketchy(int) với đối số -2.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Đây là phương thức ảo.

Do đó, chương trình sẽ xem xét xem ai đang gọi thực hiện phương thức `Sketchy()` của lớp đối tượng `B`.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Câu trả lời là: con trỏ đối tượng **pinky** đang gọi thực hiện phương thức **Sketchy()** của lớp đối tượng **B**.
- Con trỏ đối tượng **pinky** đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng **C**.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Do đó, phương thức Sketchy(int) của lớp đối tượng C được gọi thực hiện.

Trường hợp 04

```
38. class C:public B
39. {
40.     public:
41.         void Sketchy(int num)
42.         {
43.             cout<<"\n C's Sketchy("<<num<<")";
44.         }
45. };
46. void Curious(A* wacky)
47. {
48.     wacky->Sketchy();
49.     ((C*)wacky)->Sketchy(123);
50. }
```

Trong thân phương thức
`Sketchy(int)` của lớp đối tượng `C`
có duy nhất một dòng xuất liệu.

Chương trình xuất thêm một dòng
xuất liệu là:
`C's Sketchy(-2)`

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 58 được thực hiện xong.
- Chương trình xuất hai dòng xuất liệu là:
 - + B's Sketchy()
 - + C's Sketchy(-2)

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 59 đọc là con trỏ đối tượng **pinky** gọi thực hiện phương thức **Sketchy(int)** có tham số với đối số là **46**.
- Đây là phương thức ảo.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

— Do đó, chương trình xem xét xem con trỏ đối tượng **pinky** đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng nào và nó sẽ gọi thực hiện phương thức của lớp đối tượng đó.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Con trỏ đối tượng **pinky** đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng **C**.
- Do đó, phương thức **Sketchy(int)** của lớp đối tượng **C** được gọi thực hiện.

Trường hợp 04

```
38. class C:public B
39. {
40.     public:
41.         void Sketchy(int num)
42.         {
43.             cout<<"\n C's Sketchy("<<num<<")";
44.         }
45. };
46. void Curious(A* wacky)
47. {
48.     wacky->Sketchy();
49.     ((C*)wacky)->Sketchy(123);
50. }
```

Trong thân phương thức
Sketchy(int) của lớp đối tượng C
có duy nhất một dòng xuất liệu.

Chương trình xuất thêm một dòng
xuất liệu là:
C's Sketchy(46)

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 59 được thực hiện xong.
- Chương trình xuất thêm dòng xuất liệu là:
+ C's Sketchy(46)

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 60 đọc là hàm **Curious** được gọi thực hiện với đối số là **pinky**.
- **pinky** là con trỏ đối tượng thuộc lớp đối tượng **B** và đang giữ địa chỉ của đối tượng về lớp đối tượng **C**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Vào bên trong thân hàm
Curious đại diện của
pinky là wacky.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Dòng 48 đọc là con trỏ đối tượng **wacky** gọi thực hiện phương thức **Sketchy()**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Đây là phương thức ảo.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Do đó, chương trình
xem xét xem wacky là
ai.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

wacky là tham số đầu
vào của hàm Curious.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Hàm **Curious** được gọi thực hiện với đối số là **pinky**.
- **pinky** là con trỏ đối tượng thuộc lớp đối tượng **B** và đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng **C**.

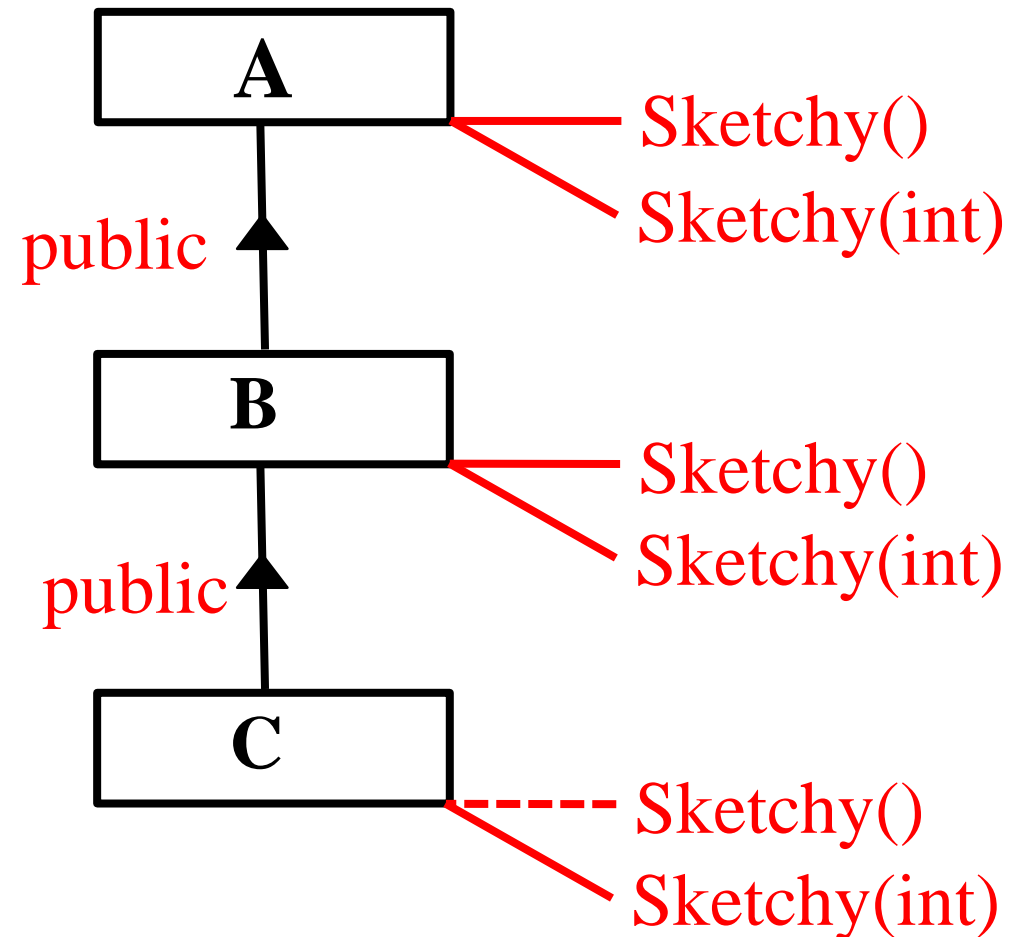
Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Do đó, phương thức **Sketchy()** của lớp đối tượng **C** được gọi thực hiện.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```



Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Mà phương thức **Sketchy()** của lớp đối tượng **C** kế thừa từ lớp đối tượng **B**, nên phương thức **Sketchy()** của lớp đối tượng **B** được gọi thực hiện

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Câu lệnh đầu tiên trong thân phương thức `Sketchy()` của lớp đối tượng `B` là câu lệnh xuất liệu.

Dòng xuất liệu là:
`B's Sketchy()`

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Trong thân phương thức Sketchy() của lớp B có lời gọi thực hiện phương thức Sketchy(int) với đối số -2.

Trường hợp 04

```
25. class B:public A
26. {
27.     public:
28.         void Sketchy()
29.         {
30.             cout<<"\n B's Sketchy()";
31.             Sketchy(-2);
32.         }
33.         void Sketchy(int num)
34.         {
35.             cout<<"\n B's Sketchy("<<num<<")";
36.         }
37. };
```

Đây là phương thức ảo.

Do đó, chương trình sẽ xem xét xem ai đang gọi thực hiện phương thức Sketchy() của lớp đối tượng B.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu trả lời là: con trỏ đối tượng **wacky** đang gọi thực hiện phương thức **Sketchy()**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu hỏi kế tiếp là wacky là ai?

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu trả lời là: **wacky** là tham số đầu vào của hàm **Curious**.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Mà hàm **Curious** được gọi thực hiện với đối số là **pinky**.
- **pinky** là con trỏ đối tượng thuộc lớp đối tượng **B** và đang giữ địa chỉ của đối tượng về lớp đối tượng **C**.

Trường hợp 04

```
25.class B:public A
26.{
27.    public:
28.        void Sketchy()
29.        {
30.            cout<<"\n B's Sketchy()";
31.            Sketchy(-2);
32.        }
33.        void Sketchy(int num)
34.        {
35.            cout<<"\n B's Sketchy("<<num<<")";
36.        }
37.};
```

Do đó, chương trình sẽ gọi thực hiện phương thức **Sketchy(int)** của lớp đối tượng **C**.

Trường hợp 04

```
38. class C:public B
39. {
40.     public:
41.         void Sketchy(int num)
42.         {
43.             cout<<"\n C's Sketchy("<<num<<")";
44.         }
45. };
46. void Curious(A* wacky)
47. {
48.     wacky->Sketchy();
49.     ((C*)wacky)->Sketchy(123);
50. }
```

Trong thân phương thức Sketchy(int) của lớp đối tượng C có duy nhất một dòng xuất liệu.

Chương trình xuất thêm một dòng xuất liệu là:
C's Sketchy(-2)

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu lệnh dòng 48 thực hiện xong.
Chương trình xuất hai dòng xuất
liệu là:

B's Sketchy()
C's Sketchy(-2)

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<");
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu lệnh dòng 49 đọc là **wacky** là con trỏ đối tượng thuộc lớp **A**, được ép kiểu thành con trỏ đối tượng thuộc lớp đối tượng **C**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu lệnh dòng 49 đọc là **wacky** là con trỏ đối tượng thuộc lớp **A**, được ép kiểu thành con trỏ đối tượng thuộc lớp đối tượng **C**.

Sau đó, gọi thực hiện phương thức **Sketchy(int)** với đối số **123**

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Đây là phương thức ảo.

Cơ chế đa xạ được ưu tiên hơn cơ chế ép kiểu.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Do đó, chương trình sẽ
xem xét xem **wacky** là ai?

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Câu trả lời là: **wacky** là tham số đầu vào của hàm **Curious**.

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Mà hàm **Curious** được gọi thực hiện với đối số là **pinky**.
- **pinky** là con trỏ đối tượng thuộc lớp đối tượng **B** và đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng **C**.

Trường hợp 04

```
38.class C:public B
39.{
40.    public:
41.        void Sketchy(int num)
42.        {
43.            cout<<"\n C's Sketchy("<<num<<")";
44.        }
45.};
46.void Curious(A* wacky)
47.{
48.    wacky->Sketchy();
49.    ((C*)wacky)->Sketchy(123);
50.}
```

Do đó, phương thức Sketchy(int) của lớp đối tượng C được thực hiện.

Trường hợp 04

```
38. class C:public B
39. {
40.     public:
41.         void Sketchy(int num)
42.         {
43.             cout<<"\n C's Sketchy("<<num<<")";
44.         }
45. };
46. void Curious(A* wacky)
47. {
48.     wacky->Sketchy();
49.     ((C*)wacky)->Sketchy(123);
50. }
```

Trong thân phương thức Sketchy(int) của lớp đối tượng C có duy nhất một dòng xuất liệu.

Chương trình xuất thêm một dòng xuất liệu là:
C's Sketchy(123)

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- Dòng 60 được thực hiện xong.
- Chương trình xuất thêm ba dòng xuất liệu là:
 - + B's Sketchy()
 - + C's Sketchy(-2)
 - + C's Sketchy(123)

Trường hợp 04

```
51. void main()  
52. {  
53.     A* inky = new B;  
54.     inky->Sketchy();  
55.     inky->Sketchy(23);  
56.     Curious(inky);  
  
57.     B* pinky = new C;  
58.     pinky->Sketchy();  
59.     pinky->Sketchy(46);  
60.     Curious(pinky);  
61. }
```

- B's Sketchy()
- B's Sketchy(-2)
- B's Sketchy(23)
- B's Sketchy()
- B's Sketchy(-2)
- B's Sketchy(123)
- B's Sketchy()
- C's Sketchy(-2)
- C's Sketchy(46)
- B's Sketchy()
- C's Sketchy(-2)
- C's Sketchy(123)

Qui tắc 01

Qui tắc 1: Con trỏ đối tượng khi gọi thực hiện một phương thức, nếu phương thức đó là phương thức không ảo thì nó sẽ gọi thực hiện phương thức của lớp đối tượng mà con trỏ thuộc về.

Qui tắc 02

Qui tắc 2: Con trỏ đối tượng khi gọi thực hiện một phương thức, nếu phương thức đó là phương thức ảo thì chương trình sẽ xem xét xem con trỏ đối tượng đang giữ địa chỉ của đối tượng thuộc về lớp đối tượng nào và chương trình sẽ gọi thực hiện phương thức của lớp đối tượng đó.

Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

TS. Nguyễn Tấn Trần Minh Khang