



# Bài 04 – Con trỏ

## TOÁN TỬ HOA THỊ - PHẦN 01

1. Hồ Thái Ngọc
2. ThS. Võ Duy Nguyên
3. TS. Nguyễn Tấn Trần Minh Khang

**Ghi nhớ**



Miền giá trị  
của một biến con trỏ là  
địa chỉ ô nhớ.

# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int* p = &a;
17.    a++;
18.    cout << "\n Gia tri tai dia chi bien ";
19.    cout << "con tro p dang giu la:" << *p;
20.    return 0;
21.}
```

# Toán tử \* – Dereferencing Operator



## — Chương trình

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int* p = &a;
17.    a++;
18.    cout << "\n Gia tri tai dia chi bien ";
19.    cout << "con tro p dang giu la:" << *p;
20.    return 0;
21.}
```

Toán tử \*: toán tử  
hoa thị (\*) được sử  
dụng để lấy và cập  
nhật giá trị tại địa chỉ  
mà biến con trỏ đang  
giữ.

# Toán tử \* – Dereferencing Operator



## — Chương trình

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    a++;
18.    cout << "\n Gia tri tai dia chi bien ";
19.    cout << "con tro p dang giu la:" << *p;
20.    return 0;
21.}
```

#1000		
#1001		
#1002		
#1003		
#1004		

Toán tử \*: toán tử  
hoa thị (\*) được sử  
dụng để lấy và cập  
nhật giá trị tại địa chỉ  
mà biến con trỏ đang  
giữ.

# Toán tử \* – Dereferencing Operator



## — Chương trình

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    a++;
18.    cout << "\n Gia tri tai dia chi bien ";
19.    cout << "con tro p dang giu la:" << *p;
20.    return 0;
21.}
```

#1000		
#1001		a
#1002		
#1003		
#1004		

Toán tử \*: toán tử  
hoa thị (\*) được sử  
dụng để lấy và cập  
nhật giá trị tại địa chỉ  
mà biến con trỏ đang  
giữ.

# Toán tử \* – Dereferencing Operator



## — Chương trình

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    a++;
18.    cout << "\n Gia tri tai dia chi bien ";
19.    cout << "con tro p dang giu la:" << *p;
20.    return 0;
21.}
```

#1000		
#1001	10	a
#1002		
#1003		
#1004		

Toán tử \*: toán tử  
hoa thị (\*) được sử  
dụng để lấy và cập  
nhật giá trị tại địa chỉ  
mà biến con trỏ đang  
giữ.

# Toán tử \* – Dereferencing Operator



## — Chương trình

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    a++;
18.    cout << "\n Gia tri tai dia chi bien ";
19.    cout << "con tro p dang giu la:" << *p;
20.    return 0;
21.}
```

#1000		
#1001	10	a
#1002		
#1003		p
#1004		

Toán tử \*: toán tử  
hoa thị (\*) được sử  
dụng để lấy và cập  
nhật giá trị tại địa chỉ  
mà biến con trỏ đang  
giữ.



# Toán tử \* – Dereferencing Operator



## — Chương trình

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    a++;
18.    cout << "\n Gia tri tai dia chi bien ";
19.    cout << "con tro p dang giu la:" << *p;
20.    return 0;
21.}
```

#1000		
#1001	10	a
#1002		
#1003	#1001	p
#1004		

Toán tử \*: toán tử  
hoa thị (\*) được sử  
dụng để lấy và cập  
nhật giá trị tại địa chỉ  
mà biến con trỏ đang  
giữ.

# Toán tử \* – Dereferencing Operator



## — Chương trình

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    a++;
18.    cout << "\n Gia tri tai dia chi bien ";
19.    cout << "con tro p dang giu la:" << *p;
20.    return 0;
21.}
```

#1000		
#1001	<del>10</del> 11	a
#1002		
#1003	#1001	p
#1004		

Toán tử \*: toán tử  
hoa thị (\*) được sử  
dụng để lấy và cập  
nhật giá trị tại địa chỉ  
mà biến con trỏ đang  
giữ.



**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**Hồ Thái Ngọc**

**ThS. Võ Duy Nguyên**

**TS. Nguyễn Tấn Trần Minh Khang**



# Bài 04 – Con trỏ

## TOÁN TỬ HOA THỊ - PHẦN 02

1. Hồ Thái Ngọc
2. ThS. Võ Duy Nguyên
3. TS. Nguyễn Tấn Trần Minh Khang

**Ghi nhớ**



Miền giá trị  
của một biến con trỏ là  
địa chỉ ô nhớ.

# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:"<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
21.    return 0;
22.}
```

# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:"<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
21.    return 0;
22.}
```

Toán tử \*: toán tử hoa thị (\*) được sử dụng để lấy và cập nhật giá trị tại địa chỉ mà biến con trỏ đang giữ.

# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:"<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
21.    return 0;
22.}
```

#1000		
#1001		
#1002		
#1003		
#1004		

Toán tử \*: toán tử hoa thị (\*) được sử dụng để lấy và cập nhật giá trị tại địa chỉ mà biến con trỏ đang giữ.



# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:"<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
21.    return 0;
22.}
```

#1000		a
#1001		
#1002		
#1003		
#1004		

Toán tử \*: toán tử hoa thị (\*) được sử dụng để lấy và cập nhật giá trị tại địa chỉ mà biến con trỏ đang giữ.

# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:"<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
21.    return 0;
22.}
```

#1000	10	a
#1001		
#1002		
#1003		
#1004		

Toán tử \*: toán tử hoa thị (\*) được sử dụng để lấy và cập nhật giá trị tại địa chỉ mà biến con trỏ đang giữ.

# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:"<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
21.    return 0;
22.}
```

#1000	10	a
#1001		
#1002		
#1003		p
#1004		

Toán tử \*: toán tử hoa thị (\*) được sử dụng để lấy và cập nhật giá trị tại địa chỉ mà biến con trỏ đang giữ.

# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:"<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
21.    return 0;
22.}
```

#1000	10	a
#1001		
#1002		
#1003	#1000	p
#1004		

Toán tử \*: toán tử hoa thị (\*) được sử dụng để lấy và cập nhật giá trị tại địa chỉ mà biến con trỏ đang giữ.

# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:<<*p;
21.    return 0;
22.}
```

#1000	10	a
#1001		
#1002		
#1003	#1000	p
#1004		

Toán tử \*: toán tử hoa thị (\*) được sử dụng để lấy và cập nhật giá trị tại địa chỉ mà biến con trỏ đang giữ.

cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:<<\*p;

# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:"<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
21.    return 0;
22.}
```

#1000	10	a
#1001		
#1002		
#1003	#1000	p
#1004		

Toán tử \*: toán tử hoa thị (\*) được sử dụng để lấy và cập nhật giá trị tại địa chỉ mà biến con trỏ đang giữ.

Giá trị tại địa chỉ con trỏ p đang giữ: 10

Giá trị tại địa chỉ con trỏ p đang giữ: 15

Giá trị tại địa chỉ con trỏ p đang giữ: 15

Giá trị tại địa chỉ con trỏ p đang giữ: 15

Giá trị tại địa chỉ con trỏ p đang giữ: 15

Giá trị tại địa chỉ con trỏ p đang giữ: 15

# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:"<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
21.    return 0;
22.}
```

#1000	<del>10</del> 15	a
#1001		
#1002		
#1003	#1000	p
#1004		

Toán tử \*: toán tử hoa thị (\*) được sử dụng để lấy và cập nhật giá trị tại địa chỉ mà biến con trỏ đang giữ.

Giá trị tại địa chỉ con trỏ p đang giữ: 15

Giá trị tại địa chỉ con trỏ p đang giữ: 15

Giá trị tại địa chỉ con trỏ p đang giữ: 15



# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:"<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
21.    return 0;
22.}
```

#1000	<del>10</del> 15	a
#1001		
#1002		
#1003	#1000	p
#1004		

Toán tử \*: toán tử hoa thị (\*) được sử dụng để lấy và cập nhật giá trị tại địa chỉ mà biến con trỏ đang giữ.

Giá trị tại địa chỉ con trỏ p đang giữ: 15

Giá trị tại địa chỉ con trỏ p đang giữ: 15

Giá trị tại địa chỉ con trỏ p đang giữ: 15

Giá trị tại địa chỉ con trỏ p đang giữ: 15

Giá trị tại địa chỉ con trỏ p đang giữ: 15

Giá trị tại địa chỉ con trỏ p đang giữ: 15



# Toán tử \* – Dereferencing Operator



— Yêu cầu: Hãy cho biết kết quả của đoạn chương trình sau?

```
11.#include <iostream>
12.using namespace std;
13.int main()
14.{
15.    int a = 10;
16.    int *p = &a;
17.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
18.    *p = 15;
19.    cout<<"\n Bien a:"<<a;
20.    cout<<"\n Giá trị tại địa chỉ con trỏ p đang giữ:"<<*p;
21.    return 0;
22.}
```

#1000	<del>10</del> 15	a
#1001		
#1002		
#1003	#1000	p
#1004		

Toán tử \*: toán tử hoa thị (\*) được sử dụng để lấy và cập nhật giá trị tại địa chỉ mà biến con trỏ đang giữ.



**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**Hồ Thái Ngọc**

**ThS. Võ Duy Nguyên**

**TS. Nguyễn Tấn Trần Minh Khang**