

# Pointer

*“Reference to a Variable”*

Prerequisite: Function

Find more contents at  
<https://sites.google.com/view/cse105june18/home>

Md. Saidul Hoque Anik  
onix.hoque.mist@gmail.com

# Understanding Pointer



Image by Marc Anderson

# Understanding Pointer



Image by Marc Anderson



Downloaded copy

# Understanding Pointer



Image by Marc Anderson



Downloaded copy  
(Modified Monalisa)



# Understanding Pointer



Image by Marc Anderson

**X** Change will not be reflected globally



Downloaded copy  
(Modified Monalisa)

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()  
{  
  
}
```

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()  
{  
    int x = 5;  
}
```

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()  
{  
    int x = 5;  
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...



# Understanding Pointer

```
#include <stdio.h>
```

```
int main()  
{  
    int x = 5;  
}
```

```
void fn(int p)  
{  
    p = p + 1;  
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(x);
    printf("x = %d \n", x);
}
```

```
void fn(int p)
{
    p = p + 1;
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(x);
    printf("x = %d \n", x);
}
```

```
void fn(int p)
{
    p = p + 1;
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...

Output

x = 5

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(x);
    printf("x = %d \n", x);
}
```

```
void fn(int p)
{
    p = p + 1;
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(x);
    printf("x = %d \n", x);
}
```

```
void fn(int p)
{
    p = p + 1;
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(x);
    printf("x = %d \n", x);
}
```

```
void fn(int p)
{
    p = p + 1;
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...



# Understanding Pointer

```
#include <stdio.h>
```

```
int main()  
{  
    int x = 5;  
    fn(x);  
    printf("x = %d \n", x);  
}
```

```
void fn(int p)
```

```
{  
    p = p + 1;  
}
```

Temporary variable, p = 5

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()  
{  
    int x = 5;  
    fn(x);  
    printf("x = %d \n", x);  
}
```

```
void fn(int p)  
{  
    p = p + 1;  
}
```

Temporary variable, p = 5

RAM Status

Address	Value
...	...
0x0101	5
...	...
0x0110	5

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(x);
    printf("x = %d \n", x);
}
```

```
void fn(int p)
{
    p = p + 1;
}
```

RAM Status

	Address	Value
	...	...
(x)	0x0101	5
	...	...
(p)	0x0110	6

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(x);
    printf("x = %d \n", x);
}
```

```
void fn(int p)
{
    p = p + 1;
}
```

RAM Status

	Address	Value
	...	...
(x)	0x0101	5
	...	...
(p)	0x0110	6

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(x);
    printf("x = %d \n", x);
}
```

```
void fn(int p)
{
    p = p + 1;
}
```

RAM Status

	Address	Value
	...	...
(x)	0x0101	5
	...	...
(p)	0x0110	6

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(x);
    printf("x = %d \n", x);
}
```

```
void fn(int p)
{
    p = p + 1;
}
```

RAM Status

	Address	Value
	...	...
(x)	0x0101	5
	...	...
(p)	0x0110	6

Output

x = 5



# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(x);
    printf("x = %d \n", x);
}
```

```
void fn(int p)
{
    p = p + 1;
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()  
{  
    int x = 5;  
    fn(x);  
    printf("x = %d \n", x);  
}
```

```
void fn(int p)  
{  
    p = p + 1;  
}
```

Original Monalisa



Modified Monalisa



RAM Status

Address	Value
...	...
0x0101	5
...	...

Output

x = 5

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(x);
    printf("x = %d \n", x);
}
```

```
void fn(int p) ← We need to send the address here
{
    p = p + 1;
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;

    printf("x = %d \n", x);
}
```

```
void fn(int * p)
{
}
```

← We need to send the **address** here  
A variable that holds address is called **Pointer**  
(Because it points to a specific location)

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;

    printf("x = %d \n", x);
}
```

```
void fn(int * p)
{
    *p = *p + 1;
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;

    printf("x = %d \n", x);
}
```

```
void fn(int * p)
{
    *p = *p + 1;
}
```

```
    a = a + 1;
```

RAM Status

Address	Value
...	...
0x0101	5
...	...



# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;

    printf("x = %d \n", x);
}
```

```
void fn(int * p)
{
    *p = *p + 1;
}
```

```
a = a + 1;
```

 Value Extraction

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

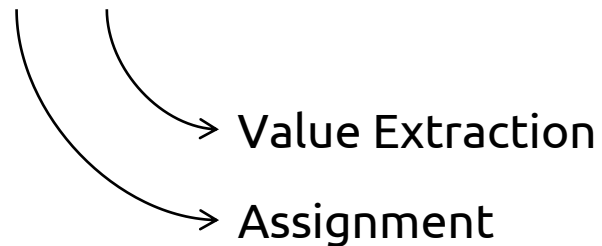
```
#include <stdio.h>
```

```
int main()
{
    int x = 5;

    printf("x = %d \n", x);
}
```

```
void fn(int * p)
{
    *p = *p + 1;
}
```

```
a = a + 1;
```



RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()  
{  
    int x = 5;  
    printf("x = %d \n", x);  
}
```

```
void fn(int * p)  
{  
    *p = *p + 1;  
}
```

Value Extraction

Assignment

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(&x);
    printf("x = %d \n", x);
}
```

```
void fn(int * p)
{
    *p = *p + 1;
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(&x); ← &x gives the address of x
    printf("x = %d \n", x);
}

void fn(int * p)
{
    *p = *p + 1;
}
```

RAM Status

Address	Value
...	...
0x0101	5
...	...

# Understanding Pointer

```
#include <stdio.h>
```

```
int main()
{
    int x = 5;
    fn(&x);
    printf("x = %d \n", x);
}
```

```
void fn(int * p)
{
    *p = *p + 1;
}
```

RAM Status

Address	Value
...	...
0x0101	6
...	...

Output

x = 6



# Benefit of Pointer

We no longer depend on return value of function

# Benefit of Pointer

We no longer depend on return value of function

```
#include <stdio.h>

int main()
{
    int x = 5;
    x = fn(x);
    printf("x = %d \n", x);
}

int fn(int p)
{
    p = p + 1;
    return p;
}
```

# Benefit of Pointer

We no longer depend on return value of function

```
#include <stdio.h>

int main()
{
    int x = 5;
    x = fn(x);
    printf("x = %d \n", x);
}

int fn(int p)
{
    p = p + 1;
    return p;
}
```

# Benefit of Pointer

How about two variables?

```
#include <stdio.h>
```

```
int fn(int p, int q)
{
    p = p + 1;
    q = q + 1;
    return p;
    return q;
}
```

 Can we write this?

# Usage (1/2)

scanf function

# Usage (1/2)

scanf function

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int x, y;
```

```
    scanf("%d%d", &x, &y);
```

```
}
```

# Usage (2/2)

swap function

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int x = 4;
```

```
    int y = 5;
```

```
}
```

```
int Swap(int * p, int * q)
```

```
{
```

```
}
```

# Usage (2/2)

swap function

```
#include <stdio.h>
```

```
int main()  
{
```

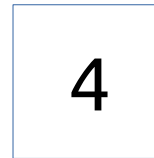
```
    int x = 4;
```

```
    int y = 5;
```

```
}
```

```
int Swap(int * p, int * q)  
{
```

```
}
```



\*p



\*q



# Usage (2/2)

swap function

```
#include <stdio.h>
```

```
int main()  
{
```

```
    int x = 4;
```

```
    int y = 5;
```

```
}
```

```
int Swap(int * p, int * q)
```

```
{
```

```
    int temp;
```

```
}
```

temp



4

\*p

5

\*q

# Usage (2/2)

swap function

```
#include <stdio.h>
```

```
int main()  
{
```

```
    int x = 4;
```

```
    int y = 5;
```

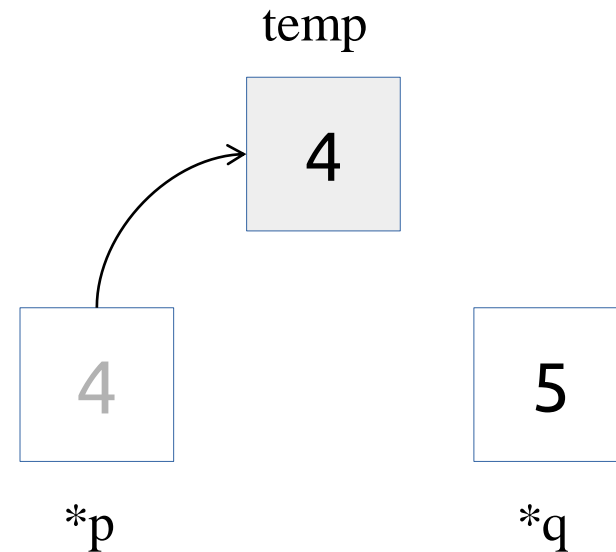
```
}
```

```
int Swap(int * p, int * q)
```

```
{
```

```
    int temp;
```

```
}
```



# Usage (2/2)

swap function

```
#include <stdio.h>
```

```
int main()  
{
```

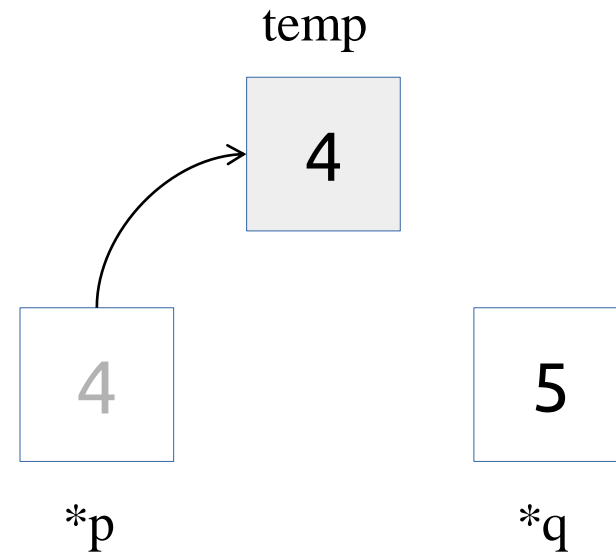
```
    int x = 4;  
    int y = 5;
```

```
}
```

```
int Swap(int * p, int * q)  
{
```

```
    int temp;  
    temp = *p;
```

```
}
```



# Usage (2/2)

swap function

```
#include <stdio.h>
```

```
int main()  
{
```

```
    int x = 4;
```

```
    int y = 5;
```

```
}
```

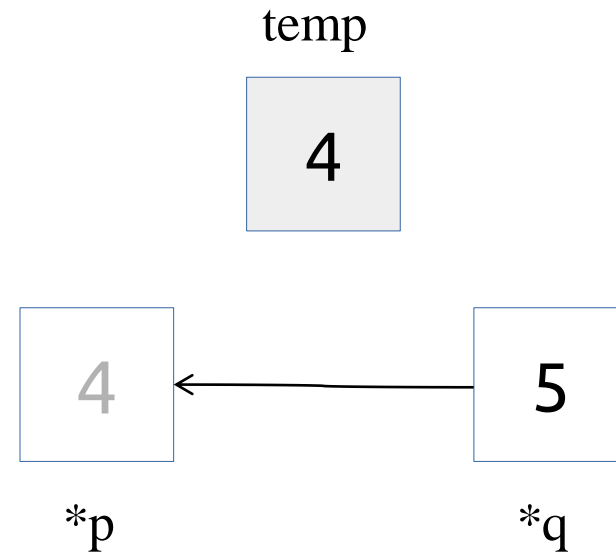
```
int Swap(int * p, int * q)
```

```
{
```

```
    int temp;
```

```
    temp = *p;
```

```
}
```



# Usage (2/2)

swap function

```
#include <stdio.h>
```

```
int main()  
{
```

```
    int x = 4;
```

```
    int y = 5;
```

```
}
```

```
int Swap(int * p, int * q)
```

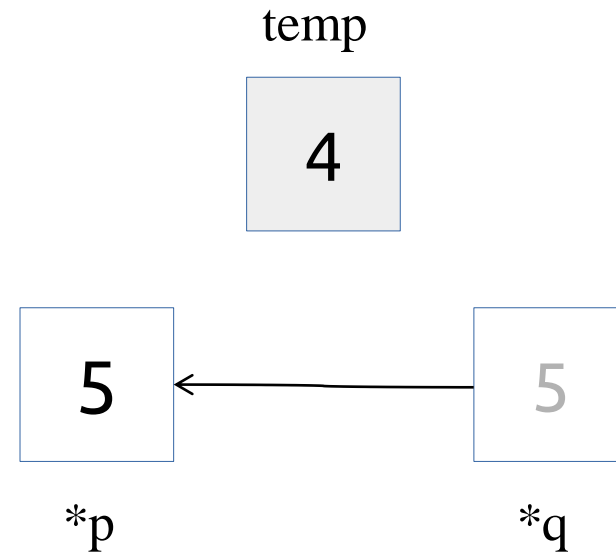
```
{
```

```
    int temp;
```

```
    temp = *p;
```

```
    *p = *q;
```

```
}
```



# Usage (2/2)

swap function

```
#include <stdio.h>
```

```
int main()  
{
```

```
    int x = 4;
```

```
    int y = 5;
```

```
}
```

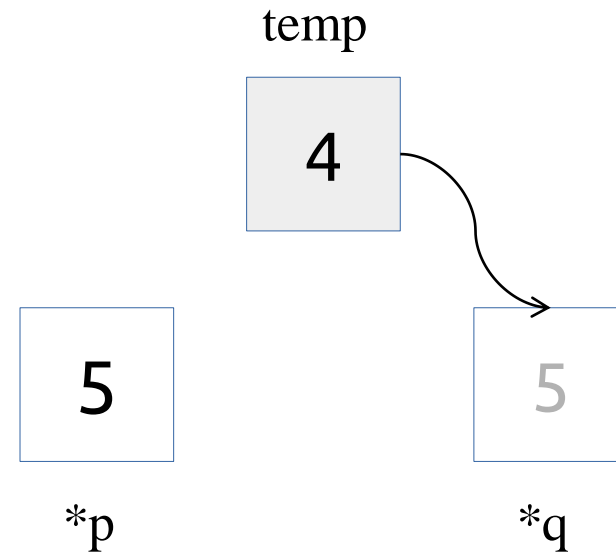
```
int Swap(int * p, int * q)  
{
```

```
    int temp;
```

```
    temp = *p;
```

```
    *p = *q;
```

```
}
```



# Usage (2/2)

swap function

```
#include <stdio.h>
```

```
int main()  
{
```

```
    int x = 4;
```

```
    int y = 5;
```

```
}
```

```
int Swap(int * p, int * q)  
{
```

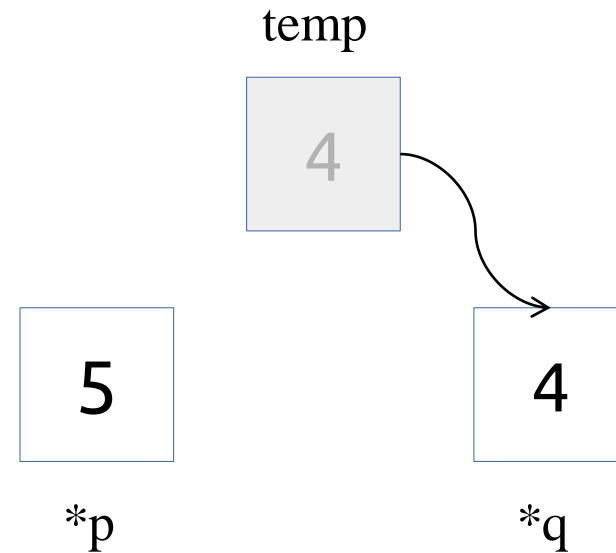
```
    int temp;
```

```
    temp = *p;
```

```
    *p = *q;
```

```
    *q = temp;
```

```
}
```



# Usage (2/2)

swap function

```
#include <stdio.h>
```

```
int main()  
{
```

```
    int x = 4;
```

```
    int y = 5;
```

```
}
```

```
int Swap(int * p, int * q)
```

```
{
```

```
    int temp;
```

```
    temp = *p;
```

```
    *p = *q;
```

```
    *q = temp;
```

```
}
```



\*p



\*q



# Usage (2/2)

swap function

```
#include <stdio.h>

int main()
{
    int x = 4;
    int y = 5;

    Swap(&x, &y);

    printf("x = %d, y = %d", x, y);
}

int Swap(int * p, int * q)
{
    int temp;
    temp = *p;
    *p = *q;
    *q = temp;
}
```

# Usage (2/2)

swap function

```
#include <stdio.h>

int main()
{
    int x = 4;
    int y = 5;

    Swap(&x, &y);

    printf("x = %d, y = %d", x, y);
}

int Swap(int * p, int * q)
{
    int temp;
    temp = *p;
    *p = *q;
    *q = temp;
}
```

Output
x = 5, y = 4