

## Preview

- Pointer variables
  - Pointer to integer
  - Pointer to character
  - Pointer to Object (will discuss later)
- Pointer and Array
  - Pointer and Integer Array
  - Pointer and Character Array
  - Array of Pointer to C-String(Array of Character)

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

1

## Pointer Variable

- A pointer variable stores the address of a variable or a structured data (or object).
- There are two important operators when working with pointers in C++: the address of **& operator** and the value of **\* operator**.
  - The **& operator** gives us the address of a variable
  - The **\* operator** gives us the value of a variable at a specified address.

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

2

## Pointer Variable

```
// pointer1.cpp
#include <iostream>
using namespace std;

int main()
{
    int x =10;

    cout <<"The Value of x is " << x <<endl;

    // display memory location of x with hexadecimal format
    cout <<"The address of x is " << &x << endl;

    /* operator provide the value when provided with a memory address
    cout <<"The value of x using * operator is " << *(&x) <<endl;

    return 0;
}
```

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

3

## Pointer Variable

- A pointer variable can be defined with the **\* operator**.
 

```
int *IntPtr;
char *ChPtr;
```
- A pointer variable name follows the same rules of composition as identifiers

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

4

## Pointer Variable

```
// pointer2.cpp
#include <iostream>
using namespace std;

int main()
{
    int x =10;
    int *xpPtr;

    xpPtr = &x; // save memory location of x to xpPtr

    cout <<"The Value of x is " << x <<endl;

    // display memory location of x with hexadecimal format
    cout <<"The address of x is " << &x << endl;

    /* operator provide the value when provided with a memory address
    cout <<"The value of x using * operator with pointer " << *xpPtr <<endl;

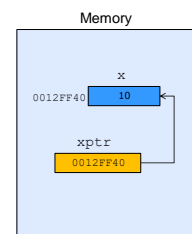
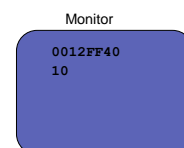
    return 0;
}
```

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

5

## Pointer Variable

```
int x =10;
int *xpPtr;
xpPtr = &x;
cout << xpPtr;
cout << *xpPtr;
```



COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

6

## Pointer Variable

```

// pointer3.cpp
// Example of pointers
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    int z = 5;
    int *y;
    int *k;
    y = &z;
    k = y;
    cout << "content of integer pointed by pointer y is "<<y <<endl;
    cout << "content of integer pointed by pointer k is "<<k << endl;
    cout << "address pointed by y is "<< y << endl;
    cout << "address pointed by k is "<< k << endl;
    cout << "address of pointer y where an address pointed by y is saved "<< &y <<endl;
    cout << "address of pointer k where an address pointed by k is saved "<<&k <<endl;
    *y = 10;
    cout << "content of integer pointed by pointer y is changed to "<< z << endl;
    return 0;
}
    
```

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park 7

```

int z = 5;
int *y;
int *k;
y = &z;
k = y;
cout << *y<<endl;
cout << *k<<endl;
cout << z <<endl;
cout << y <<endl;
    
```

Memory

Monitor screen

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park 8

## Pointer Variable

```

// pointer4.cpp
//Example of integer pointer
#include <iostream>
using namespace std;
int main()
{
    int i =10;
    int *x = &i;
    int *y;
    y = &i;
    cout <<"The pointer x is stored at the memory address " << &x <<endl;
    cout <<"The pointer x stored the memory address of i: " << x <<endl;
    cout <<"The value of i accessed through pointer x is " << *x <<endl;

    *x = *x + 1; // value pointed by the pointer x is increments by 1
    cout <<"i (though pointer) = " << *x <<endl;
    cout <<"i (direct access) = " << i <<endl;
    cout <<"The size of integer pointer x is " << sizeof *x <<endl;
    cout << sizeof(x) <<" bytes."<<endl;

    return 0;
}
    
```

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park 9

## Pointer Variable

(Pointer and Name of Array)

- Name of Array – Save a memory location where first element of the array is located
- A pointer - Save a memory location where an type of value is located

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park 10

## Pointer Variable

(Pointer and Name of Array)

```

// pointer5.cpp
// pointer and name of array
#include <iostream>
using namespace std;
int main()
{
    int A[3] = {1, 2, 3};
    int *APtr;
    APtr = A;

    // print element of array
    for (int i = 0; i <3; i++)
        cout << A[i]<<endl;

    for (int i = 0; i<3; i++)
        cout << *(A+i) <<endl;

    // print element of array with a pointer with * operator
    for (int i = 0; i<3; i++)
        cout << *(APtr+i) <<endl;

    // print element of array with a pointer
    for (int i = 0; i<3; i++)
        cout << APtr[i] <<endl;

    return 0;
}
    
```

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park 11

## Pointer Variable

(Pointer and Array of Integer)

```

int A[3] = {1, 2, 3};
int *APtr;
APtr = A;
cout << A[0] << *A << endl;
cout << A[1] << *(A+1)<< endl;

cout << APtr[0] << *APtr << endl;
cout << APtr[1] << *(APtr+1)<< endl;
    
```

Memory

Monitor

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park 12

## Pointer Variable

(Pointer and Array of Character)

```
// pointer6.cpp: Pointer and Array Example
#include <iostream>
using namespace std;

int main()
{
    char MyName[12] ="Sang-Eon";
    char *ChPtr;
    // display the string Sang-Eon
    cout << MyName<<endl;
    // display the first element of the character array
    cout << *MyName <<' ' << MyName[0] <<endl;
    //display the second element of the character array
    cout << *(MyName+1) <<' ' << MyName[1] <<endl;

    ChPtr = MyName;

    // display the string Sang-Eon
    cout << MyName<<endl;
    // display the first element of the character array
    cout << *ChPtr <<' ' << ChPtr[0] <<endl;
    //display the second element of the character array
    cout << *(ChPtr+1) <<' ' << ChPtr[1] <<endl;
    return 0;
}
```

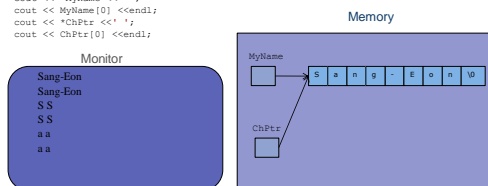
COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

13

## Pointer Variable

(Pointer and Array of Character)

```
char MyName[12] ="Sang-Eon";      cout << *(MyName+1) <<' ' << MyName[1] <<endl;
char *ChPtr;                    cout << *(ChPtr+1) <<' ' << ChPtr[1] <<endl;
ChPtr = MyName;
cout << MyName<<endl;
cout << ChPtr<<endl;
cout << *MyName <<' ' << MyName[0] <<endl;
cout << *ChPtr <<' ' << ChPtr[0] <<endl;
cout << ChPtr[0] <<endl;
```



COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

14

## Pointer Variable

(Pointer and Array of Character)

```
// pointer7.cpp :Pointer and Array Example
#include <iostream>
using namespace std;

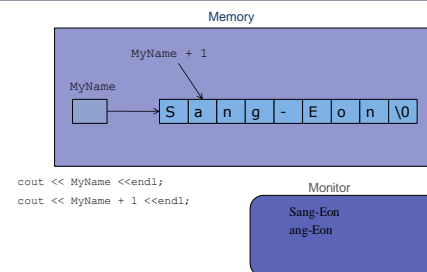
int main ()
{
    char MyName[12] ="Sang-Eon";
    // display a string referenced by MyName
    cout << MyName <<endl;
    // display a string referenced by MyName +1
    cout << MyName + 1 <<endl;
    return 0;
}
```

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

15

## Pointer Variable

(Pointer and Array of Character)



```
cout << MyName <<endl;
cout << MyName + 1 <<endl;
```

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

16

## Pointer Variable

(Pointer and Array of Character)

```
// pointer8.cpp: Pointer and Array Example
#include <iostream>
using namespace std;

int main()
{
    char *Name ="Sang-Eon";

    cout << Name << endl; // display Sang-Eon
    // display the first character of the string
    cout << Name[0]<<' ' << *Name <<endl;

    //display the second character of the string
    cout << Name[1]<<' ' << *(Name+1)<<endl;
    return 0;
}
```

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

17

## Pointer Variable

(Array of Pointers to C-string (array of character))

```
//pointer9.cpp: Example array of pointer
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    const char *point[4]= {"you", "are", "my", "sunshine"};
    cout << point[0]<< endl;
    cout << point[1]<< endl;
    cout << point[2]<< endl;
    cout << point[3]<< endl;
    return 0;
}
```

COSC 220 Computer Science II, Fall 2020  
Dr. Sang-Eon Park

18

### Pointer Variable

(Array of Pointers to C-string (array of character))

```
const char *point[4] = {"you", "are", "my", "sunshine"};
```

Memory

0012FF34 → y o u \0  
 0012FF38 → a r e \0  
 0012FF3C → m y \0  
 0012FF40 → s u n s h i n e \0

COSC 220 Computer Science II, Fall 2020  
 Dr. Sang-Eon Park 19

### Pointer Variable

(Array of Pointers to C-string (array of character))

```
const char *point[4] = {"you", "are", "my", "sunshine"};
cout << point[0] << endl;
cout << point[1] << endl;
cout << point[2] << endl;
cout << point[3] << endl;
```

Memory

0012FF34 → y o u \0  
 0012FF38 → a r e \0  
 0012FF3C → m y \0  
 0012FF40 → s u n s h i n e \0

Monitor  
 you  
 are  
 my  
 sunshine

COSC 220 Computer Science II, Fall 2020  
 Dr. Sang-Eon Park 20

### Pointer Variable

(Array of Pointers to C-string (array of character))

```
const char *point[4] = {"you", "are", "my", "sunshine"};
cout << point[0]+2 << endl;
cout << point[1]+2 << endl;
cout << point[2]+1 << endl;
cout << point[3]+3 << endl;
```

Memory

0012FF34 → y o u \0  
 0012FF38 → a r e \0  
 0012FF3C → m y \0  
 0012FF40 → s u n s h i n e \0

Monitor  
 ou  
 e  
 y  
 shine

COSC 220 Computer Science II, Fall 2020  
 Dr. Sang-Eon Park 21

### Pointer Variable

(Array of Pointers to C-string (array of character))

```
const char *point[4] = {"you", "are", "my", "sunshine"};
cout << *point[0] << endl;
cout << *point[1] << endl;
cout << *(point[2]+1) << endl;
cout << *(point[3]+2) << endl;
```

Memory

0012FF34 → y o u \0  
 0012FF38 → a r e \0  
 0012FF3C → m y \0  
 0012FF40 → s u n s h i n e \0

Monitor  
 y  
 a  
 y  
 u

COSC 220 Computer Science II, Fall 2020  
 Dr. Sang-Eon Park 22

### Pointer Variable

(Array of Pointers to C-string (array of character))

```
const char *point[4] = {"you", "are", "my", "sunshine"};
cout << point << endl;
cout << point +1 << endl;
cout << point +2 << endl;
cout << point +3 << endl;
```

Memory

0012FF34 → y o u \0  
 0012FF38 → a r e \0  
 0012FF3C → m y \0  
 0012FF40 → s u n s h i n e \0

Monitor  
 0012FF34  
 0012FF38  
 0012FF3C  
 0012FF40

COSC 220 Computer Science II, Fall 2020  
 Dr. Sang-Eon Park 23