

9.

Pointer



경희대학교
KYUNG HEE UNIVERSITY

● 실습 목표 - 포인터의 개념 이해와 활용

1. 포인터의 선언과 초기화
2. 매개변수 전달 방식에 따른 값의 변화와 포인터의 값 읽고 쓰기
 - Call by Value
 - Call by Reference
 - Call by Address(Pointer)
3. 포인터 반환 함수의 구현과 응용

● 실습 문제 1 < Section 9-1 ~ 9-5 >

/** 목표 : 포인터 변수 기본 사용법을 익히고 , 정수형 포인터 변수 값과 정수형 변수가 저장된 메모리 주소값의 비교를 통해 포인터의 의미를 이해 **/

고객프로그래밍 과목의 성적 3개를 정수로 입력 받고 출력하는 프로그램을 작성하고 자 한다. 아래 각 요구 사항에 대하여 main() 을 작성하라.

- 정수형 변수 3개 선언: grade1, grade2, grade3
- 정수형 변수를 가리키는 포인터 변수 3개 선언 : pgrade1, pgrade2, pgrade3
- 포인터 변수 초기화 : ex) pgrade1 = &grade1
- 성적입력 : ex) cin >> grade1
- 성적출력 : 변수 값, 변수의 주소값, 포인터 변수 값, 포인터 변수의 주소값 출력 및 비교

```
Ex) cout << "Value : grade1 = " << setw(8) << grade1 << " *pgrade1 = " << setw(8) << *pgrade1  
    << " *&grade1 = " << setw(8) << *&grade1 << endl;  
    cout << "Addr : &grade1 = " << &grade1 << " pgrade1 = " << pgrade1  
    << " &pgrad1 = " << &pgrade1 << endl;
```

/** 변수명, 참조 변수, 참조 주소 연산자 표현 방식을 이해하라.

● 실습 문제 2 < Section 9-5 >

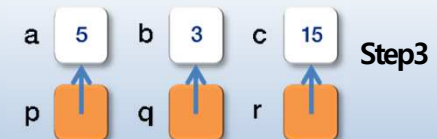
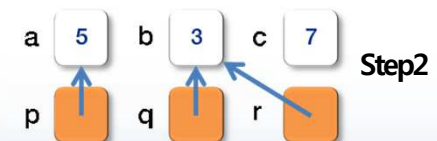
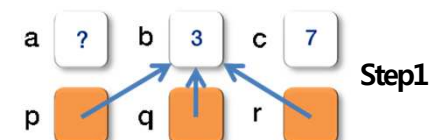
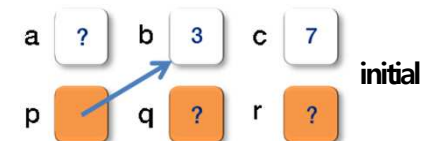
/** 목표 : 포인터 사용법을 익히고, 세 가지 형식(변수명, 참조 변수, 참조 주소 연산자) 간 덧셈을 통해 동작 방법을 이해한다. **/

아래 그림과 같이 정수형 변수 3개와 그것을 가리키는 포인터 변수 3개를 조작하는 Main() 프로그램을 작성하라.

```
int main() {
    /** 오른쪽 그림(initial)과 같이 정수형 변수 a, b, c와 정수형 포인터 변수 p, q, r인 선언되어 있다.
        각 변수는 아래와 같이 초기화 되어있다. **/
    int a;   int b = 3;   int c = 7;   int* p = &b;   int* q;   int* r;

    /** 오른쪽 그림(Step1~Step3)을 참조하여 포인터 변수를 조작하는 코드를 작성하라. **/
    // step1 code
    // step2 code
    // step3 code ( p,q,r을 이용하여 변수 c에는 a,b,c의 합을 넣어라.)

    cout << "a = " << *p << "(" << p << ")" << endl;
    cout << "b = " << *q << "(" << q << ")" << endl;
    cout << "c = " << *r << "(" << r << ")" << endl;
    return 0;
}
```



● 실습 문제 3 < Section 9-6 >

/** 목표 : 함수에서 하나 이상의 변수를 인자를 보낼 때, 매개변수의 형식에 따라 값이 변하는 동작 방식을 이해하라. */

아래 그림과 같이 두 개의 정수형 변수를 입력 받아 그 값을 서로 바꾸는 함수(swap) 아래 요구사항에 맞춰 3가지 함수를 구현하고 아래 main() 내 테스트 코드를 통해 그 결과를 확인하라.

- 정수형 변수 a = 5, b = 7 로 초기화
- swapValue(int x, int y) 구현(input : 두 개의 정수형 변수, post: x와 y의 값 변경)
- swapRef(int& x, int& y) 구현(input: 두 개의 정수형 참조 변수, post: x와 y의 값 변경)
- swapAddr(int* px, int* py) 구현(input: 두 개의 포인터 변수, post: px와 py가 가리키는 값 변경)

```
int main()
{
    int x = 5, y = 7; int *px = &x; int *py = &y;
    swapValue(x, y); cout << "swapValue()함수 호출 후 " << setw(5) << x << setw(5) << y << endl;
    swapRef(x, y);   cout << "swapRef()함수 호출 후  " << setw(5) << x << setw(5) << y << endl;
    swapAddr(px, py); cout << "swapAddr()함수 호출 후 " << setw(5) << x << setw(5) << y << endl;
    swapAddr(&x, &y); cout << "swapAddr()함수 호출 후 " << setw(5) << x << setw(5) << y << endl;
    return 0;
}
```

output

swap1()	함수 호출 후	5	7
swap2()	함수 호출 후	7	5
swap3()	함수 호출 후	5	7
swap3()	함수 호출 후	7	5

● 실습 문제 4 < Section 9-6 >

/** 목표 : 포인터를 반환하는 함수를 통해 포인터 활용법을 익힌다. */

실습 1번의 프로그램에 아래 요구사항에 맞는 topGrade() 함수를 구현하고 main() 내 테스트 코드를 통해 그 결과를 확인하라.

// input : 세 개의 정수형 포인터 변수(성적값)

// output : 가장 큰 값을 갖는 변수의 포인터

- int* topGrade (int* px, int* py, int*pz)

int main()

{ // 실습1 코드(중략)

int* top; // 가장 높은 값을 갖는 변수의 주소 저장 위한 포인터 변수

top = topGrade(pgrade1, pgrade2, pgrade3); // topGrade() 함수 호출

// 반환된 주소값(top)과 실습 1의 출력 결과물에서 세 변수 주소값과 비교

cout << "Top = " << *top << "(addr : " << top << ")" << endl; return 0;

}

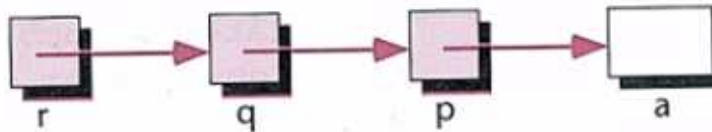
output

```
55 95 80
Value : grade1 =      55 *pgrade1 =      55 *&grade1 =      55
Addr  : &grade1 = 0039FEE8 pgrade1 = 0039FEE8 &pgrad1 = 0039FEC4
Value : grade2 =      95 *pgrade2 =      95 *&grade2 =      95
Addr  : &grade2 = 0039FEDC pgrade2 = 0039FEDC &pgrad2 = 0039FEB8
Value : grade3 =      80 *pgrade3 =      80 *&grade3 =      80
Addr  : &grade3 = 0039FED0 pgrade3 = 0039FED0 &pgrad3 = 0039FEAC
Top = 95(addr : 0039FEDC)
```


● 실습 문제 5 < Section 9-7 >

/** 목표 : 포인터의 포인터 사용방법을 이해한다. */

아래 그림과 같이 정수형 포인터 변수 p, q, r 이 정수형 변수 a를 가리키도록 main() 내 코드를 작성하고 그 출력값을 확인하라.



```
int main() {  
    int a; int* p; int** q; int*** r;  
    p = &a; q = &p; r = &q;  
    cout << "Enter a number: ";  
    // (code) 변수 a를 이용하여 입력받기  
    cout << "Your number is : " << a << endl;  
    cout << "\nEnter a number: ";  
    // (code) 포인터 변수 p를 이용하여 입력받기  
    cout << "Your number is : " << a << endl;  
    cout << "\nEnter a number: ";  
    // (code) 포인터 변수 q를 이용하여 입력받기  
    cout << "Your number is : " << a << endl;  
    cout << "\nEnter a number: ";  
    // (code) 포인터 변수 r를 이용하여 입력받기  
    cout << "Your number is : " << a << endl; }  
}
```

output

```
Enter a number: 1  
Your number is : 1  
  
Enter a number: 2  
Your number is : 2  
  
Enter a number: 3  
Your number is : 3  
  
Enter a number: 4  
Your number is : 4
```

● 실습 문제 6 < Section 9-7 >

/** 목표 : 포인터의 포인터 활용을 이해한다. */

정수형 변수를 가리키는 포인터 값을 변경하는 같은 기능의 두 함수 swapDptr , swapRefPtr 함수를 구현하고, main() 내 테스트 코드를 이용하여 그 결과를 확인하라.

// input : 정수형 변수를 가리키는 포인터의 주소(포인터)

void swapDptr(int** x, int** y) // 두 포인터의 주소 변경

// input : 정수형 참조 변수의 포인터

void swapDptr(int*& x, int*& y) // 두 포인터의 주소 변경

output

```
int main(){
    int x = 5, y = 7;
    int *px = &x; int *py = &y;
    cout << "swapDptr()함수 호출 전 " << px << " " << py << " " << x << " " << y << endl;
    swapDptr(&px, &py);
    cout << "swapDptr()함수 호출 후 " << px << " " << py << " " << x << " " << y << endl;
    swapRefPtr(px, py);
    cout << "swapRefPtr()함수 호출 후 " << px << " " << py << " " << x << " " << y << endl;
    return 0;
}
```

swapDptr()	함수 호출 전	0024FE14	0024FE08	5	7
swapDptr()	함수 호출 후	0024FE08	0024FE14	5	7
swapRefPtr()	함수 호출 후	0024FE14	0024FE08	5	7

● 실습 문제 7 < Section 9-8 >

/ 목표 : 포인터의 종류를 살펴보고, 그 사용법을 이해하자**/**

아래 코드를 작성하여 그 컴파일 에러문을 확인하고, 올바르게 수정하여라. 또한 출력 결과가 가지는 의미를 생각해 보자.

```
int main() {
    int i = 0x12345678;    int* pi = &i;
    char c = 'C';          char* pc = &c;
    float f = 700.5f;      float* pf = &c;
    bool b = true;         bool* pb = &b;
    short int s = 456;     short int* ps = &i;

    pc = (char*)&i; // char* 타입의 포인터로 i 를 가리킨다.

    // 다음 출력의 의미를 생각해 보자.
    cout << hex << "*pc = " << (int) *pc << endl; // int 로의 형변환은 문자가 아닌 숫자로 출력하기 위함.
    return 0;
}
```

error

```
); error C2440: 'initializing' : cannot convert from 'char *' to 'float *'
;ast or function-style cast
); error C2440: 'initializing' : cannot convert from 'int *' to 'short *'
;ast or function-style cast
```

output

```
*pc = 78
계속하려면 아무 키나 누르십시오 . . .
```