

교학습내용

- ♀ 12.2 동적 메모리 사용 절차 (2)
- **♀** 12.3 동적 메모리 사용 예제 (1)



할 학습목표

- 12.2 동적 메모리 할당 함수 사용법을 익힌다.
- 12.3 예제를 통해 동적 메모리 할당 사용법을 익힌다.





C프로그래밍및실습





- 1. 동적 할당 개요
- 2. 동적 메모리 사용 절차 (2)
- 3. 동적 메모리 사용 예제
- 4. 기타 동적 메모리 할당 함수



⑤ 동적 할당 기본 예제

● 동적으로 메모리를 할당하고 해제하는 라이브러리 함수는 (stdlib.h) 헤더 파일에 선언

```
#include <stdlib.h> // 동적 메모리 관련 함수 사용을
                // 위한 헤더
int main() {
  int *p = NULL; // 동적 메모리 접근을 위한 포인터 변수
  p = (int *) malloc( 5*sizeof(int) ); // 동적 메모리 할당
  p[0] = 1; // 동적 메모리 사용: 배열 형태
  *(p+2) = 3; // 동적 메모리 사용: 포인터 형태
  free(p);
             // 동적 메모리 해제
  return 0;
```



■ 동적 메모리 해제 : free() 함수

• 지정된 메모리 공간을 해제(free)

함수 원형	<pre>void free(void * ptr);</pre>	
함수 인자	해제할 메모리 공간을 가리키는 포인터 변수	
기능	ptr이 <mark>가리키는</mark> 메모리 해제	

≫ 주의!! ptr 변수 자체를 해제시키는 것 아님

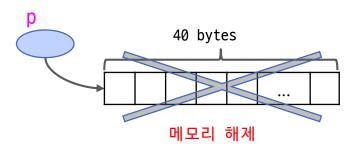
• 예시

```
int *p = NULL;

p = (int *) malloc( 10*sizeof(int) );

...

free( p ); // p가 가리키는 영역 해제
```





※실습하기



[예제 12.1] 다음 지시대로 동적 메모리 할당 및 해제하는 코드를 작성해보자.

- 1. 정수 한 개를 저장할 수 있는 메모리 공간 할당하고 해제하기
- 2. float형 실수 한 개를 저장할 수 있는 메모리 공간 할당하고 해제하기
- 3. double형 실수 15개를 저장할 수 있는 메모리 공간 할당하고 해제하기
- 4. 다음과 같은 구조체 한 개를 저장할 수 있는 메모리 공간 할당하고 해제하기

```
struct student{
  int id; char name[8]; double grade;
};
```

동적 메모리를 <u>안전</u>하게 사용하기 위한 주의사항

- malloc()의 반환 값을 검사하여 **메모리 할당의 성공 여부 확인**
 - ✓ NULL 포인터 반환하는 경우 : 요청한 메모리 크기만큼 연속된 메모리 공간을 확보하지 못할 때
- 비슷하게, 해제하려는 메모리 주소가 NULL인지 검사

```
int *p = NULL;
p = (int *) malloc( 10*sizeof(int) );
if(p == NULL) { // 메모리 할당 실패하면,
  printf("Not enough memory!"); // 오류 상황 알리고,
                          // 함수 종료
  return -1;
if( p != NULL ) // p가 NULL이 아닌 경우에만
  free(p);
              //free() 함수 호출
```

🗊 메모리 할당 방식 비교

	정적 메모리 할당	동적 메모리 할당
필요한 메모리 크기 결정시점	프로그램 작성 단계	프로그램 실행 중
메모리 할당 및 해제 시점	시스템이 자동으로 할당 및 해제	개발자가 명시적으로 할당 및 해제 함수 호 출
프로그램 실행 중 메모리 크기 변경 여부	불가	가능
메모리 누수(leak) 가능성	없음	있음
메모리 사용의 효율성	비효율적	효율적

C프로그래밍및실습





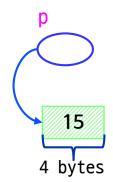
- 동적 할당 개요
- 2. 동적 메모리 사용 절차 (2)
- 3. 동적 메모리 사용 예제
- 4. 기타 동적 메모리 할당 함수

3. 동적 메모리 할당 프로그램 예제

1) 동적으로 메모리를 할당 받아 사용하는 기본 프로그램

하나의 정수를 저장하는 메모리를 동적 할당 받아 사용 (실용성은 없음)

```
#include <stdio.h>
#include <stdlib.h> // 동적 할당 함수가 선언된 헤더 파일
int main() {
  int *p = NULL;
                           // 포인터 변수 선언
  p = (int *)malloc(sizeof(int)); // 동적 할당
  if (p == NULL) { // 동적 할당 오류 검사
     printf("Not enough memory!");
     return -1;
  *p = 15;
                         // 동적 할당 메모리에 값 저장
  printf("동적 메모리 할당 (정수형): %d", *p);
  free(p);
  return 0;
```



3. 동적 메모리 할당 프로그램 예제

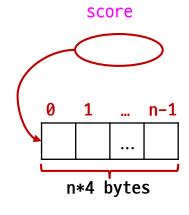


💷 2) 일차원 배열을 동적 할당 받아 사용하는 프로그램

- 여러 개의 정수를 저장하는 메모리를 동적 할당 받아 사용
- 문제: 한 학생의 과목별 점수를 입력 받아 평균 점수를 구한 후 출력하는 프로그램을 작성하시오.
 - ① 프로그램 실행 중에 사용자로부터 과목 수 n을 직접 입력 받음
 - ② n개의 정수형 값을 저장할 수 있는 메모리를 동적으로 할당 받아, 할당 받은 메모리의 시작 주소를 score에 저장
 - ③ 할당 받은 메모리를 score[0]~score[n-1]까지 일반 배열 원소처럼 참조하여 사용
 - ④ 할당 받은 메모리 해제

3. 동적 메모리 할당 프로그램 예제

```
int n, i, sum = 0;
int *score = NULL;
scanf("%d", &n);
                        // ① 과목 수 입력받기
score = (int *) malloc( n*sizeof(int) ); // ② 배열 동적 할당
if (score == NULL) {
                           // 동적 할당 오류 검사
for (i=0; i<n; i++) {
  scanf("%d", &score[i]); // ③ 과목별 점수 입력받기
  sum += score[i]; // 누적 합 계산
printf("%.1f", (double)sum/n); // 평균 계산
free(score);
                        // ④ 동적 할당 메모리 해제
return 0;
```





학습 정리

- free() 함수는 할당된 메모리를 해제하여 시스템에 되돌려 줌
- <mark>정적 할당</mark>은 할당 메모리가 코드 작성 단계에서 정해지고, 프로그램을 실행할 때마다 동일한 크기로 할당하는 방식
- **동적 할당**은 프로그램 실행 단계에서 결정되는 크기에 따라 메모리를 할당하는 방식
- 동적 할당과 포인터를 활용하여 **일차원 배열**을 사용할 수 있음(예제2)

