

C Programming Part. 2

1. 기억 클래스
2. 포인터
3. 구조체
4. 파일처리
5. 자료구조와 알고리즘

2학기 프로그래밍 수행 평가 계획(60% 중간20% 기말20%)

실습(60점)

1. 포인터 및 정렬/검색 알고리즘(20점)
2. 구조체(20점)
3. 파일처리(20점)

보고서 및 프로젝트(40점)

1. 보고서 및 평소 과제 제출(20점)
2. 개별 프로젝트(20점)
구조체 및 파일 입출력을 이용한 작품 제출

기억 클래스(Storage class)

자료가 저장되는 **장소**와 **생존기간(life time)**,
유효범위(scope)를 규정하는 것.

형식

[기억 클래스] 데이터형 변수명;

11. 기억 클래스(Storage class)

auto

static

extern

register

auto (자동 범위 변수)

- 함수나 블록 내부에 선언
- 선언된 함수나 블록 내에서만 유효한 지역 변수
- 생성 : 변수가 선언된 함수나 블록이 호출 또는 진입될 때
- 소멸 : 호출된 함수나 블록의 실행이 종료될 때
- 초기화 : 실행(run-time) 시 초기화
 - 1) 초기화 값을 지정하지 않으면 쓰레기 값으로 초기화
 - 2) 초기화 값을 지정한 경우, 함수나 블록이 호출될 때마다 초기값이 항상 새로 부여.

static (정적 변수)

- 함수나 블록 내부에 선언 : 내부 정적 변수
 - 선언된 함수나 블록 내에서만 유효한 정적 변수
- 함수의 외부에 선언 : 외부 정적 변수
 - 선언된 이후부터 프로그램 끝까지 유효한 정적 변수
- 생성 : 컴파일 시 생성
- 소멸 : 프로그램 종료 시 소멸
- 초기화 : 컴파일(compile-time) 시 단 1회만 초기값이 부여.
 - 1) 초기화 값을 지정하지 않으면 초기값이 자동 부여
 - 2) 초기화 값을 지정한 경우, 초기화한 값 부여

```
#include <stdio.h>

void sub_fun(void);

int main(){
    int i;
    for(i=0;i<5;i++)
        sub_fun();
    return 0;
}

void sub_fun(void){
    auto int x=0;
    static int y=0;
    printf("auto = %d, static = %d\n",x,y);
    x++;
    y++;
}
```

```
#include <stdio.h>

void sub_fun(void);
static int x;

int main(){
    static int y;
    x++;
    y++;
    z++;
    return 0;
}

void sub_fun(void){
    static int z;
    x++;
    y++;
    z++;
}
```


extern (외부 변수)

- 다른 프로그램 모듈에서도 접근 가능한 변수
- 독립적으로 컴파일 된 여러 개의 프로그램 파일에서 값을 공유하여 참조해야 할 경우 사용
- 함수 외부에 전역변수로 기억 클래스 없이 정의, 외부 변수를 참조하고자 하는 모듈에서 extern을 사용하여 선언하며 사용 가능
- 초기화 : 컴파일 시 단 1회만 초기값이 부여.
 - 1) 외부 변수가 선언된 모듈에서만 초기화 가능
 - 2) 초기화 값을 지정하지 않으면 초기값이 자동 부여
- 동일한 이름의 지역 변수와 전역 변수가 중복된 경우 지역 변수가 우선된다.

```
#include <stdio.h>

void sub_fun(void);
int a=10, b=20;

int main(){
    extern c;
    printf( " a=%d, b=%d, c=%d\n " ,a,b,c);
    sub_fun();
    return 0;
}
int c=30;
void sub_fun(void){
    printf( " a + b + c=%d\n " ,(a+b+c));
}
```

[참고] 함수에 대한 기억 클래스

- static 또는 extern
- 디폴트 클래스는 **extern**
 - 프로그램을 구성하는 모든 파일에서 사용 가능
- **static 함수**는 정의된 파일에서만 사용 가능
 - 비공개 함수로 사용될 수 있다.

main.c

```
#include <stdio.h>
#include <stdlib.h>
void func1(void){
    printf("func1\n");
}
int main(int argc, char *argv[])
{
    func1();
    func2();
    func3();
    system("PAUSE");
    return 0;
}
```

func2.c

```
static void func2(void){
    printf("func2\n");
}
```

func3.c

```
extern void func3(void){
    printf("func3\n");
}
```

11. 기억 클래스

선언장소	기억 클래스	지정자	기억장소	생존 기간	유효범위
함수의 외부	외부변수	정의 : 없음	메모리 (정적영역)	영구적	프로그램전역
		선언 : extern			Static
	외부정적변수	내부정적변수			
함수의 내부	레지스터변수	Register	레지스터		
	자동변수	Auto	메모리 (스택)		

[보고서 : C언어에서 사용하는 메모리 구조]

스택(Stack), 힙(Heap), 데이터 영역 등을 조사하고
메모리의 각 영역에
변수의 종류 별로
저장되는 내용을 정리해 보세요.

제출 파일명 :

학번_이름_C언어에서 사용하는메모리구조.hwp

[C언어에서 사용하는 메모리 구조]

데이터영역	스택 영역	힙 영역
정적(static) 변수 전역변수	자동변수 매개변수	동적 할당 변수

```
void ff(int j);  
int main(){  
    auto int i;  
    for(i=1 ; i <=3 ; i++)  
        ff(i);  
    return 0;  
}
```

```
void ff(int j)  
{  
    static int a = 100;  
    auto int k = 1;  
    ++k;  
    printf("%d + %d + %d = %d \n", a, k, j, a+k+j);  
    a += 10;  
}
```

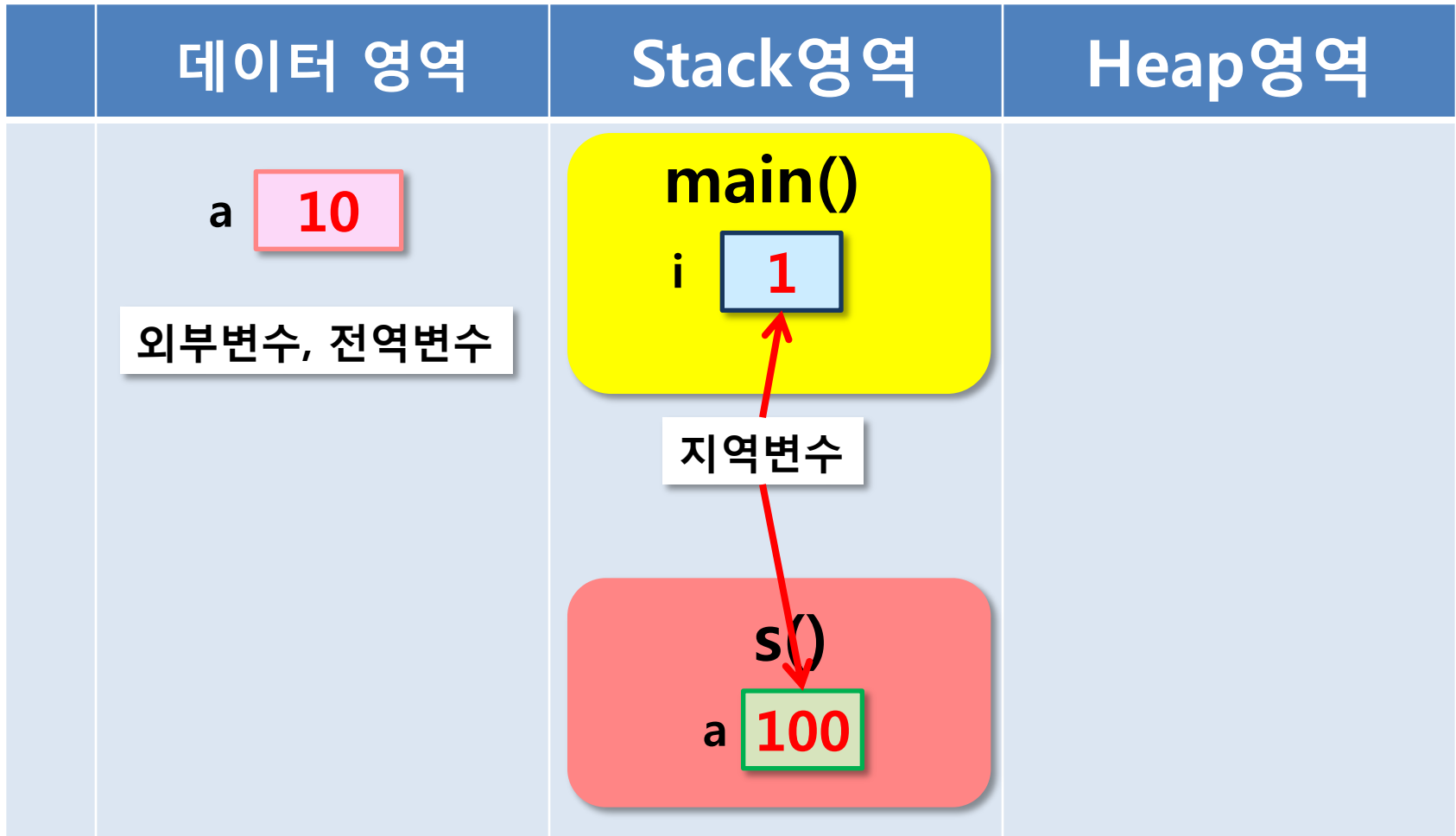

08.기억클래스.hwp(1~8번까지)의 코드에 있는 변수들을 각각 영역별로 정리

(아래 그림은 1번 문제에 대한 예시)

	데이터 영역	Stack영역	Heap영역
	<div>a100</div>	<div>main()<div>i1</div><div>ff()<div>j1</div><div>k1</div></div></div>	

```
extern int a;
void s(void);
int main(){
    int i;
    for(i=1; i <= 3; ++i){
        ++a;
        printf("%d in main,",a);
        s();
    }
    return 0;
}
```

```
int a=10;
void s(void){
    int a=100;
    ++a;
    printf("%d in subfunction.\n", a);
}
```



```
int x;
```

```
void sub1(), sub2();
```

```
int main(){
```

```
    int i;
```

```
    sub1();
```

```
    sub2();
```

```
    printf("In main, i=%d, x=%d \n", i, x);
```

```
    return 0;
```

```
}
```

```
static int i;
```

```
void sub1(){
```

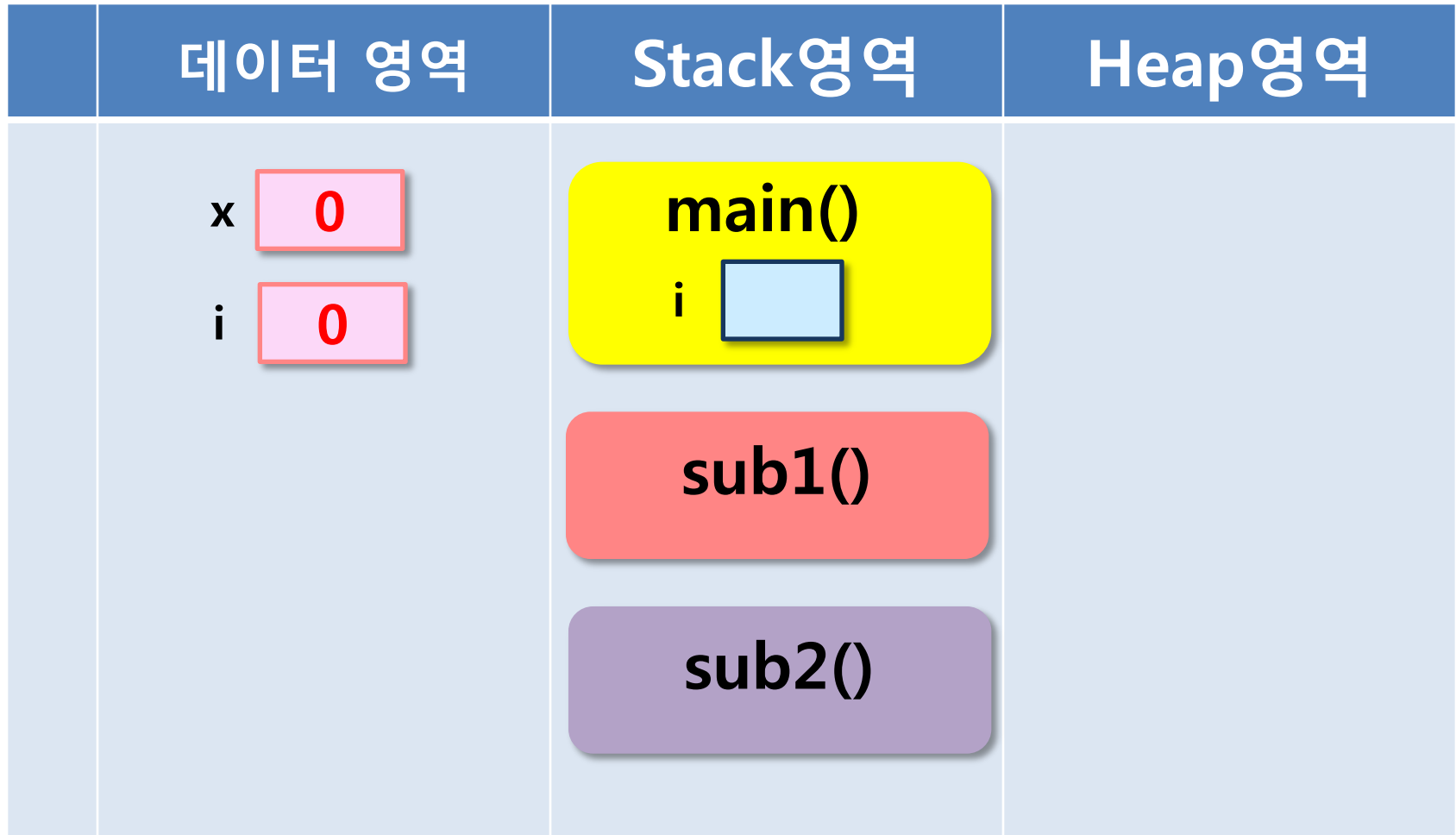
```
    printf("In sub1, i=%d, x=%d \n", ++i, ++x);
```

```
}
```

```
void sub2(){
```

```
    printf("In sub2, i=%d, x=%d \n", ++i, ++x);
```

```
}
```



```
void main(){  
    int n=5;  
    printf("n=%d\n",n);  
    {  
        int n=10;  
        printf("n=%d\n",n);  
        n++;  
    }  
    printf("n=%d\n",n);  
}
```

```
void sub(void){  
    auto int x=0;  
    static int y=0;  
    printf("auto x=%d, static y=%d\n",x,y);  
    x++, y++;  
}
```

```
void main(){  
    int i;  
    for(i=0; i<5; i++)  
        sub();  
}
```

```
int fx=0;
void func(void){
    static int sx=0;
    int ax=0;
    printf("%d%d%d\n", ax++, sx++, fx++);
}
int main(void){
    int i;
    for(i=0;i<5;i++)
        func();
    return 0;
}
```



```
int a=10, b=20;
```

```
void sub_fun();
```

```
void main(){
```

```
    extern c;
```

```
    printf("a=%d, b=%d, c=%d\n",a,b,c);
```

```
    sub_fun();
```

```
}
```

```
int c=30;
```

```
void sub_fun(){
```

```
    printf("a+b+c=%d\n",a+b+c);
```

```
}
```

```
int x=700;
```

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

```
int main(void){  
    int i;  
    int x=800;  
    print_x();  
    printf("x=%d\n",x);  
    for(i=0;i<5;i++){  
        int x=i*100;  
        printf("%d\n",x);  
    }  
    printf("x=%d\n",x);  
    return 0;  
}
```

1 100 + 2 + 1 = 103
110 + 2 + 2 = 114
120 + 2 + 3 = 125

2 11 in main, 101 in subfunction
12 in main, 101 in subfunction
13 in main, 101 in subfunction

3 In sub1, i=1, x=1
In sub2, i=2, x=2
In main, i=쓰레기값, x=2

4 n=5
n=10
n=5

5 auto x=0, static y=0
auto x=0, static y=1
auto x=0, static y=2
auto x=0, static y=3
auto x=0, static y=4

6 000
011
022
033
044

7 a=10, b=20, c=30
a+b+c=60

8 x=700
x=800
0
100
200
300
400
x=800