

Java를 알고 C배우기

# 컴퓨터프로그래밍3

## week 4-1 함수와 스택-스택프레임

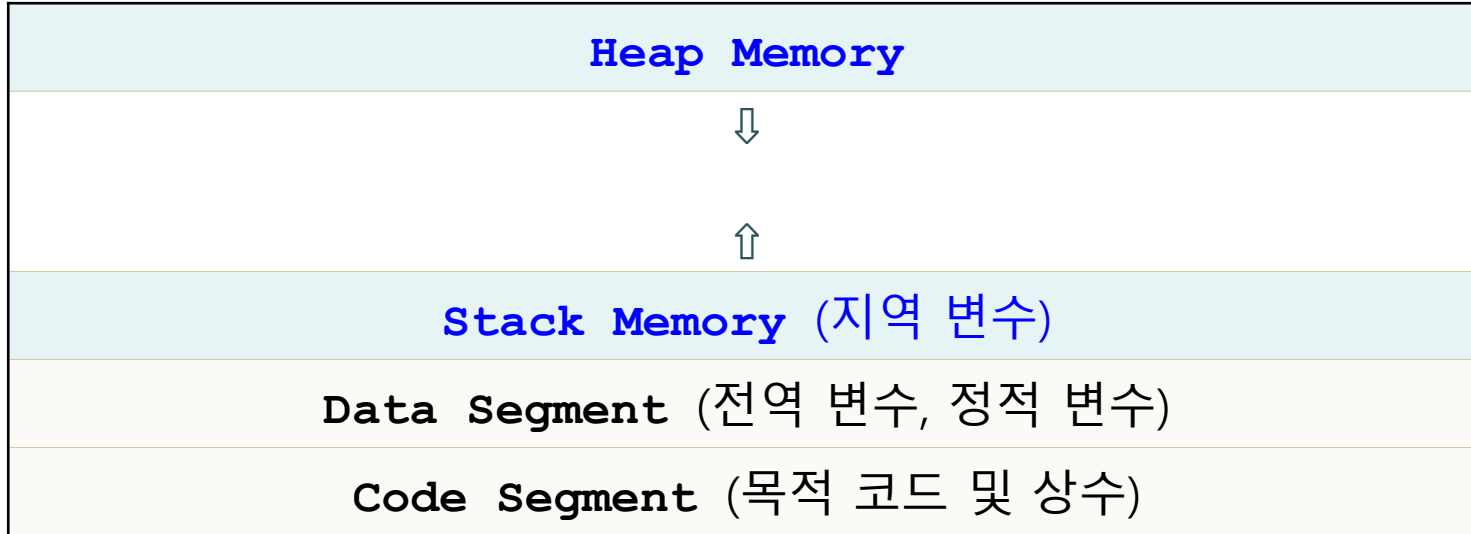
2022.1학기  
충남대 조은선

# 스택 Stack

- ▶ 스택
  - ▶ 나중에 입력된 것이 먼저 접근되는 자료구조 (LIFO)
  - ▶ 연산 PUSH, POP, TOP
  - ▶ 예?
- ▶ 함수와 스택의 연관성?
  - ▶ 나중에 호출된 함수가 먼저 리턴됨



# 메모리 모습



- ▶ 프로그램 실행 중의 메모리 구조
- ▶ 스택이 위로 자라고 힙이 아래로 자란다.(또는 그 반대)

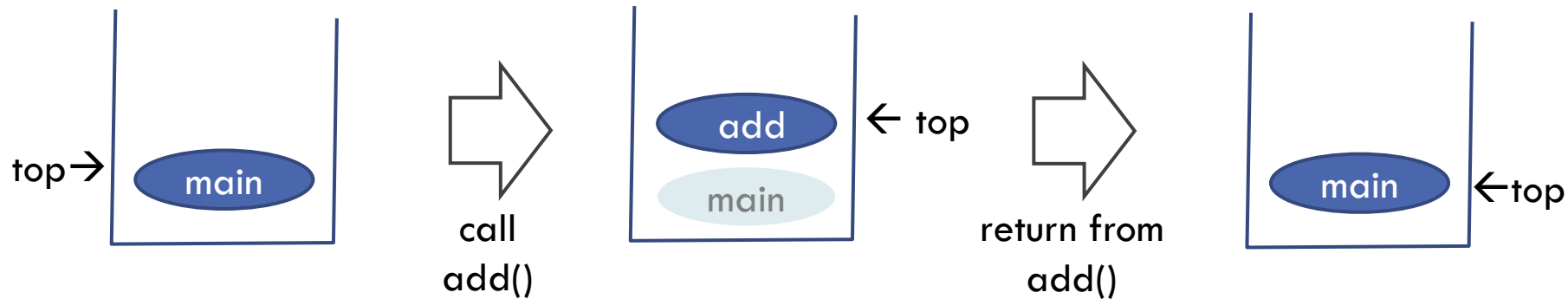
참고: “스택오버플로우”

# 스택 프레임 (Stack Frame)

- ▶ 함수를 호출할 때마다 스택 메모리에 **스택 프레임**이란 것을 push
- ▶ **활성화 레코드(Activation Record)**라고도 부름
- ▶ **스택 프레임 내용**
  - ▶ **매개 변수(Parameter)**  
호출 함수가 전달한 인자 값
  - ▶ **지역 변수(Local Variables)**  
함수 내에서 선언한 변수
  - ▶ **리턴 주소(Return Address)**  
함수 실행을 마친 다음 실행할 명령문의 주소
  - ▶ **리턴 값(Return Value)**  
호출 함수에게 돌려줄 값

# 스택과 함수 호출 관계

- ▶ main에서 add()를 호출하고 다시 main으로 리턴할 때
  - ▶ add가 실행되면 add의 스택프레임을 push
  - ▶ add 실행이 끝나면 **add의 스택 프레임을 pop**
  - ▶ 이제 그 아래 깔려 있는 **main의 스택 프레임이 스택 top에** 위치
  - ▶ main의 스택 프레임이 **다시 활성화**



# 메모리 모습 예

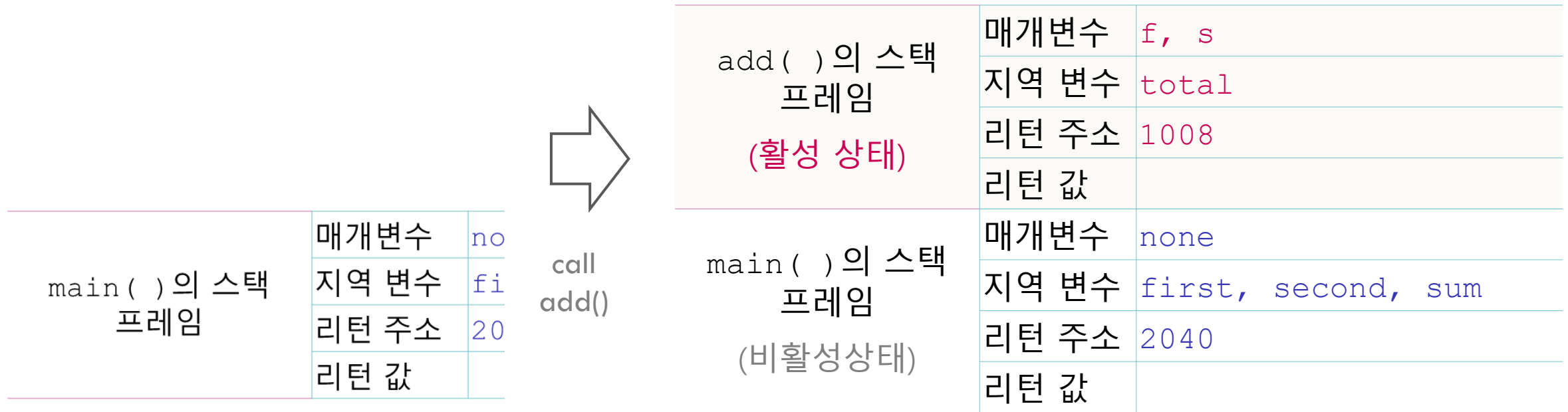
|                               |       |        |  |      |
|-------------------------------|-------|--------|--|------|
| main( )의<br>스택 프레임<br>(활성 상태) | 매개변수  |        |  |      |
|                               | 지역 변수 | first  |  |      |
|                               |       | second |  |      |
|                               |       | sum    |  |      |
|                               | 리턴 주소 |        |  | 2040 |
|                               | 리턴 값  |        |  |      |

- ▶ 운영체제가 main을 호출
- ▶ main의 스택 프레임을 푸쉬

...

|         |                                     |   |
|---------|-------------------------------------|---|
| 코드 세그먼트 |                                     | <pre> int main( ){     int first, second, sum;     printf("Enter two input integers.\n");     scanf("%d%d", &amp;first, &amp;second);     sum = add(first, second);     printf("The sum is %d.\n", sum);     return 0; } </pre> |
|         | <p>1004</p> <p>1008</p> <p>1016</p> | <pre> int add(int f, int s){     int total;     total = f + s;     return total; } </pre>   |

# 메모리 모습 예 (1004번지 수행)-add 호출



## ▶ main이 add를 호출

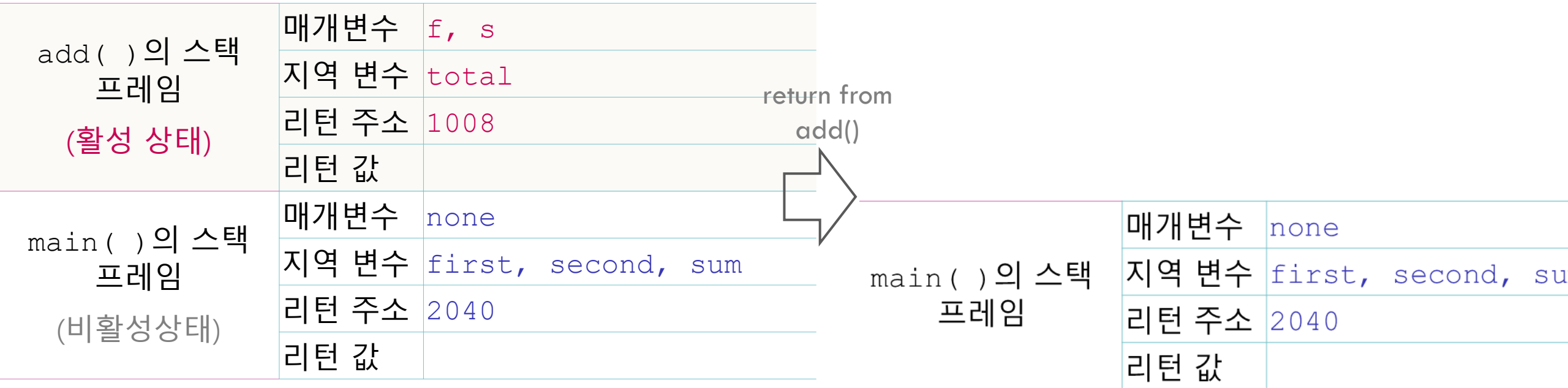
- ▶ main의 스택 프레임만 있는 스택에 add의 스택 프레임을 push
- ▶ 리턴 주소 1008은 add 실행을 끝내고 되돌아갈 main의 주소
- ▶ (코드 세그먼트는 불변)

```

int main( ){
    int first, second, sum;
    printf("Enter two input\n");
    scanf("%d%d", &first, &second);
    1004 sum = add(first, second);
    1008 printf("The sum is %d.\n", sum);
    return 0;
}

1016 int add(int f, int s){
    int total;
    total = f + s;
    return total;
}
    
```

# 메모리 모습 예 (1004번지 수행)-add 리턴



## ▶ 스택프레임의 역할

- ▶ **스택프레임은 함수 호출** 이전 상태를 기록
- ▶ Callee 함수를 호출하기 이전에 현재 함수의 현재까지 상태를 기록
- ▶ Callee 함수로부터 리턴된 다음에는, 호출한 함수의 이전 상태를 복원

```

int main( ){
    int first, second, sum;
    printf("Enter two input\n");
    scanf("%d%d", &first, &second);
    1004 sum = add(first, second);
    1008 printf("The sum is %d.\n", sum);
    return 0;
}

1016 int add(int f, int s){
    int total;
    total = f + s;
    return total;
}
    
```



# 스택 프레임과 변수

- ▶ 매 순간, **스택 top에 있는 하나의 스택 프레임만 활성화**
  - ▶ add를 실행하는 동안에는 add의 스택 프레임 안에 있는 f, s, total에만 접근 가능
  - ▶ main의 스택 프레임 안에 있는 first, second, sum에는 접근 불가능
  - ▶ 스택프레임이 현재 실행되고 있는 함수의 지역 참조 환경을 형성함
- ▶ 스택 프레임의 크기는 **컴파일 타임에 결정**
  - ▶ 매개변수, 지역 변수의 개수와 타입을 고려
- ▶ 지역변수의 초기화 이전 쓰레기값
  - ▶ 이전에 pop된 스택프레임에 있었던 값

# Quiz

- ▶ 다음 중 거리가 먼 것을 모두 고르시오.
  - (1) 함수의 지역변수는 스택프레임에 저장된다.
  - (2) 스택프레임은 리턴 주소를 포함한다.
  - (3) 함수의 호출이 일어나면 호출된 함수의 스택프레임이 스택 메모리에 push된다.
  - (4) 함수의 코드는 스택프레임과 함께 스택에 저장된다.
  - (5) 호출된 함수가 리턴될 때 top에 있는 스택프레임이 pop 된다.
  - (6) 스택메모리 top에 있는 스택프레임의 지역 변수만 접근 가능하다