

# Coding

## Dangling Pointer

```
int main(void)
{
    int * pInt = (int*)malloc(sizeof(int));
    *pInt = 10;
    free(pInt);
    *pInt = 10; // <- Dangling Pointer Issue
    return 0;
}
```

- 동적 할당된 메모리 영역을 **Free**시켜 해제된 메모리 공간을 가리키고 있는 **pInt**의 포인터 변수를 **Dangling Pointer**라고 한다.
  - 문법 에러는 발생하지 않지만, 런 타임 중 **Free**한 위치에 새로 동적 할당이 이루어져 중요한 정보가 저장되어 버린다면 심각한 문제가 발생할 수 있다!!

```
int main(void)
{
    int * pInt = (int*)malloc(sizeof(int));
    *pInt = 10;
    free(pInt);
    pInt = NULL;
    *pInt = 10; // <- Dangling Pointer Issue
    return 0;
}
```

- 해제된 포인터 변수에 **NULL**을 입력하면 런타임 중 해제된 메모리 접근 시 “처리되지 않은 예외가 throw됨: 쓰기 액세스 위반입니다. **pInt**이(가) **nullptr**였습니다.”라는 에러 메시지를 출력한다.
-