

프로그래밍 언어론	6주차 과제
소프트웨어학과 2학년	2017038064 김동용

1. Stack-dynamic variable 와 heap-dynamic variable의 공통점과 차이점을 서술하시오.

공통점은 런타임 시에 allocation이 일어난다.

차이점은 Stack-dynamic의 경우 lifetime은 해당 변수의 선언을 포함한 program unit이 실행될 때부터 종료될 때 까지이며 Heap-dynamic의 경우 lifetime은 사용자가 allocation할 때부터 deallocation할 때 까지입니다.

2. 다음 C프로그램에서 n의 scope와 lifetime을 설명하시오, n의 경우 호출한 함수를 같이 명시하시오(예 : fac(3) 때 선언된 n의 scope는..이고 lifetime은..이다). 추가적으로 lifetime의 경우 각 n이 allocation 및 dealloction이 되는 시간을 순서대로 나열하시오.

예) fac(1)의 n이 allocation→fac(2)의 n이 dealloction→....

```
void main( ){
    int f;

    f=fac(3);
}

int fac(int n){

    if(n>1) return n*fac(n-1);

    Else return 1;

}
```

fac(3) 때 선언된 n의 scope는 fac(3)함수 안에서 visible하며 lifetime은 n=3에서 fac(3)이 실행될 때부터 종료될 때 까지입니다. fac(2) 때 선언된 n의 scope는 fac(2)함수 안에서 visible하며 lifetime은 n=2에서 fac(2)이 실행될 때부터 종료될 때 까지입니다. fac(1) 때 선언된 n의 scope는 fac(1)함수 안에서 visible하며 lifetime은 n=1에서 fac(1)이 실행될 때부터 종료될 때 까지입니다.

또한, lifetime의 경우, fac(3)의 n이 allocation→fac(2)의 n이 allocation→fac(1)의 n이 allocation→fac(1)의 n이 deallocation→fac(2)의 n이 deallocation→fac(3)의 n이 deallocation입니다.

3. 다음 1. 2. 3의 composite type들을 Cartesian product, mapping, disjoint union, recursion 등을 이용해 나타내시오.

```
enum Suit {club, diamond, heart, spade};  
1. struct Card {Suit s; byte r;};  
2. typedef Card [100] Hand;  
3. struct Turn {bool pass; Card play;};
```

```
1.struct Card {Suit s; byte r;};
```

cartesian product

→Card = Suit X byte = {club, diamond, heart, spade} X {-128, ..., 127}

```
2. typedef Card[100] Hand;
```

mapping

→ Index = {0 ... 99}

Card = Suit X byte = {club, diamond, heart, spade} X {-128, ..., 127}

Hand = Index → Card = {0, 1, 2, ..., 99} → {{club, 1}, {diamond, 1}, ... }

```
3. struct Turn {bool pass; Card play;};
```

cartesian product

→Card = Suit X byte = {club, diamond, heart, spade} X {-128, ..., 127}

Turn = bool X Card= {true, false} X {(club, 1), (diamond, 1), (heart, 1), (spade, 1) ...}

4. 아래 C 예시에서 x=y가 에러를 일으키는 원인을 C언어의 local variable의 특징을 이용하여 설명하시오.

```
Int main( )
```

```
{  
    Int x[10];  
    Int y[10];  
    x=y;  
    return 0;  
}
```

Local variable의 scope는 program unit안에서 선언되고, 해당 unit안에서 visible합니다. Lifetime은 함수가 실행될 때부터 종료될 때까지 지속됩니다.

코드 상에서 x는 메인 함수 안에 포함된 local variable이며 x를 선언한 시점에서 int type의 원소 10개를 배열로 allocation하였고 x=y에서 deallocation되지 않은 x를 재사용하려고 시도했기 때문에 에러가 발생합니다.