

Week 4-1 : Inline Functions, Default Arguments, Function Overloading

Part1. Inline Functions

프로그램 코드라인 안으로 들어간 함수.

함수의 인라인화를 통해 성능을 향상시킬 수 있음.

자료형에 독립적이지 못함.

inline 키워드에 대한 처리는 컴파일러에 의해서 처리됨. 컴파일러가 함수의 인라인화가 오히려 성능에 해가 된다고 판단할 경우, 키워드를 무시하기도 하며, 필요한 경우 일부 함수를 임의로 인라인 처리하기도 한다.

```
#include <iostream>

inline int multiplier(int x)
{
    return x*x;
}

int main(void)
{
    std::cout<<mutiplier(10)<<std::endl;
    return 0;
}
```

Part2. Default Arguments

case1.

```
int number(int num=0)
{
    return num;
}
```

인자를 전달하지 않으면 0이 전달된 것으로 간주하며, 0이 default 값이 된다. 때문에 함수 number()과 number(0)의 호출은 같은 결과를 출력한

다.

case2.

```
int adder(int num1=0, int num2=0)
{
    return num1+num2;
}
```

인자를 전달하지 않으면, 각각 0, 0의 default값을 가지며, 때문에 함수 adder()과 adder(0,0)은 같은 결과를 출력한다.

case3.

```
int adder(int num1, int num2=0)
{
    return num1+num2;
}
```

adder함수를 다음과 같이 변경했을 때, adder(10)과 adder(10,20)의 결과 값은 각각 10과 30이 출력된다.

case4.

```
int adder(int num1=0, int num2)
{
    return num1+num2;
}
```

adder함수를 이와 같이 변경하였을 때, 제대로 수행되지 않는다. 왜냐하면, 전달되는 인자가 왼쪽에서부터 채워지기 때문이다.

[Exercise]

Make a 'Vending' class, which has 'cup' variable(private) and 'coffee' method (parameter : kinds of coffee, money). When you first, just input kinds of coffee and show the result (default money value), Second input money value and show the number of cups of coffee result.

- Price (use #define)

americano : 200, caffe-latte : 400

- Handling error case

1. input a number over '3' value in kinds of coffee

2. short of money

```
CA. C:\windows\system32
**just select** <1:americano, 2:caffe-latte>
2
short of money
0 cups of coffee
**input money**
2000
5 cups of coffee
계속하려면 아무 키나 누르십시오 . . .
```

```
CA. C:\windows\system32
**just select** <1:americano, 2:caffe-latte>
3
Not resistered goods
0 cups of coffee
[error] Not resistered goods
계속하려면 아무 키나 누르십시오 . . .
```