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

Part1. Inline Functions
Function which is located in program code line
Inline Function can improve performance depend on Data Type
Compiler handle inline keyword. When compiler think inline is not good for performance, then ignore inline keyword. sometimes compiler makes inline arbitrarily.

Part2. Default Arguments

case1.

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

When user passes nothing to parameter, parameter has 0 value , 0 is default value. so, number() and number(0) show same result.

case2.

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

When user passes nothing to parameter, parameter has 0, 0 values respectively regard this situation as 0 value passed, 0 is
default value. so, adder() and adder(0,0) show same result.

case3.

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

When you change adder function like this, adder(10) and adder(10,20) show 10, 30 result values.

case4.

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

When you change adder function like this, does not work. because, parameter be filled from left to right.
[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