**CPP 기말고사 정리**

Week 09.

CPP에서 지원하는 String을 사용하지 않고 str 카피하기.

myStrCopy(char\* changeStr, char\* mainStr){

1. For 문 사용하기  
   int k;  
   for (k=0; mainStr[k] != ‘\0’; k++)  
    changeStr[k] = mainStr[k];  
   changeStr[k+1] = ‘\0’
2. 포인터 변수 접근법  
   while(\*mainStr == NULL){  
    \*changeStr = \*mainStr;  
    mainStr++;  
    changeStr++;  
   }  
   changeStr = NULL;

}

CPP에서 지원하는 String을 사용하고 카피하기.

1. 변수 선언시 ()를 이용해 copy할 객체나 선언할 str 타입을 넣는다.
2. Operator 오버로딩을 통해 선언한다.

String.substr(n,m) => index값의 n부터 m까지를 출력한다.

String.at(index) => index에 있는 값을 출력한다.

String.size() => 문자열 길이를 리턴한다.

String.capacity() => 현재 string object에 할당된 메모리 값을 리턴한다.

String.front()/.back() => 문자열의 처음과 끝을 리턴한다.

Week 10.

getline(cin, 변수, 구분:default=’\n’) => 유저의 cin(키보드입력)을 변수에 대입한다. 구분을 만나면 입력을 멈춘다.

String.reserve(int) => int개의 문자를 담을 수 있는 공간을 예약 신청한다.

#include <iostream>

using namespace std;

class Digit

{

private:

int m\_digit;

public:

Digit(int digit = 0) : m\_digit(digit) {}

friend ostream & operator << (ostream &out, const Digit &d) // 전역 함수

{

out << d.m\_digit;

return out;

}

friend Digit & operator ++ (Digit & digit) // ⭐전역 함수

{

++digit.m\_digit;

return digit;

}

};

int main()

{

Digit d(5);

cout << ++d << endl; // 6 출력

cout << d << endl; // 6 출력

return 0;

}

friend Digit operator ++ (Digit & digit, int) // 전역 함수 구현시

{

Digit temp(digit.m\_digit);

++digit;

return temp;

}

Digit operator ++ (int) // 멤버 함수 구현시

{

Digit temp(m\_digit);

++(\*this);

return temp;

}