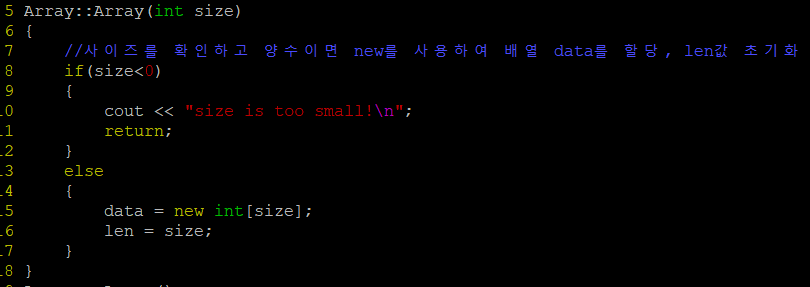
3주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20192135 이름: 윤영인

**1.**

1) Array::Array(int size)



배열의 크기인 size를 입력 받아 Array 인스턴스를 생성하는 함수이다. 음수일 경우, 에러 메시지를 출력하고 함수를 종료하며, 양수일 경우, new로 size 만큼 int형 메모리를 할당하여 int형 포인터인 data에 넣어준다. len에는 배열의 크기인 size를 넣어준다.

2) Array::~Array()

텍스트이(가) 표시된 사진

자동 생성된 설명

생성자로 생성한 인스턴스를 소멸시키는 소멸자로, delete를 이용하여 동적할당으로 할당한 메모리를 해제한다.

3) int Array::length() const

텍스트이(가) 표시된 사진

자동 생성된 설명

배열의 크기를 반환하는 멤버 함수로, 생성자 멤버 함수에서 배열의 크기를 넣어준 len을 리턴한다.

4) int& Array::operator[](int i)

텍스트, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명

배열에 원소를 대입하거나 값을 반환하는 [] 연산자의 오버로딩이다. 즉, left value의 용도이므로, a[2] = 4이면 a[2]에 4를 넣어준다. 따라서 배열의 인덱스가 범위 내에 있을 경우, 배열에 원소 값을 넣어주고 그 값을 리턴한다. 그렇지 않으면 에러 메시지를 출력한 후 static int형인 tmp를 리턴한다.

5) int Array::operator[] (int i) const

텍스트, 실내이(가) 표시된 사진

자동 생성된 설명

배열의 원소 값을 반환하는 멤버 함수이다. 배열의 인덱스가 범위 내에 있으면 해당 원소 값을 리턴하고, 그렇지 않으면 에러 메시지를 출력하고 함수를 종료한다.

6) void Array::print()

텍스트이(가) 표시된 사진

자동 생성된 설명

배열의 모든 내용을 출력하는 멤버 함수로, for문을 돌면서 배열의 범위 내의 원소 값을 출력한다.

7) Makefile

rangearray.cpp, RangeArray.h, main.cpp, Array.h, array.cpp 로 실행 파일인 main을 생성하기 위해 Makefile을 작성한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

컴파일하는 도구로 “g++ -c”를 사용하기 위해, CC와 CFLAG의 매크로를 정의하고, 실행파일의 이름을 main으로 하기 위해 TARGET 매크로를 정의한다. 오브젝트 파일의 이름을 main.o array.o rangearray.o로 하여 하나로 묶는 매크로인 OBJS를 정의한다. make all을 입력하면 all : $(TARGET) 문장에 의해 TARGET 파일이 생성되고 이 파일을 생성하기 위해 OBJS 매크로에 해당하는 파일에 의존한다. “$(CC) -o $@ $^” 명령어에 의해 의존 목록 파일 전체에 대응하여 TARGET 이름으로 실행 파일을 생성한다. main.o array.o rangearray.o의 파일을 생성하기 위해 “$(CC) $(CFLAG) -o $@ $<” 명령어를 사용하여 -c -o 옵션으로 .c파일과 같은 이름의 .o 파일을 생성하고, 각 .o 파일들이 의존하는 헤더를 명시한다. “make clean”을 입력하면 “rm $(OBJS) $(TARGET)”에 의해 오브젝트 파일과 실행파일이 지워지며, 파일 중에 clean이라는 파일이 있어 make clean이 수행되지 않는 것을 방지하기 위해 .PHONY를 정의한다.

**2.**

1) String.h – Class Str을 정의하여 헤더 파일에 넣어주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

#ifndef ~ #endif을 이용하여 \_STRING\_H를 #define 문장으로 정의하지 않았다면 #endif 이전의 내용을 include 영역에 포함하도록 한다. 클래스 내부에서만 접근이 가능하도록 private 필드에 string의 내용과 길이에 해당하는 변수를 넣어주었다.

2) Str::Str(int leng)

텍스트, 모니터, 실내, 화면이(가) 표시된 사진

자동 생성된 설명

배열의 크기인 leng을 입력 받아 Str 인스턴스를 생성하는 생성자 멤버 함수이다. leng이 음수이면 에러 메시지를 출력하고 함수를 종료하고, 그렇지 않으면 leng 만큼 char형의 메모리를 할당하여 str이라는 char형 포인터에 넣어준다. 배열의 크기를 나타내는 변수인 len에 leng을 넣어준다.

3) Str::Str(char \*neyong)

텍스트, 모니터, 화면, 실내이(가) 표시된 사진

자동 생성된 설명

초기화할 문자열을 입력 받아 Str 인스턴스를 생성하는 생성자 멤버 함수이다. strlen으로 neyong이 비어 있는지 확인하여, 비어 있으면 에러 메시지를 출력하고 종료하고, 비어 있지 않으면, neyong의 길이를 strlen으로 재어 len에 넣어준다. new를 이용하여 len만큼 char형 메모리를 할당하여 char형 포인터인 str에 넣어준다. 메모리 공간이 할당된 str에 neyong의 내용을 넣기 위해 strcpy로 str에 neyong의 내용을 복사한다.

4) Str::~Str()

텍스트이(가) 표시된 사진

자동 생성된 설명

인스턴스를 소멸하는 소멸자 멤버 함수로, delete를 이용하여 str을 할당 해제한다.

5) int Str::length(void)

텍스트이(가) 표시된 사진

자동 생성된 설명

str의 길이를 리턴하는 멤버 함수로, 생성하면서 길이를 넣어주었던 len을 리턴한다.

6) char \*Str::contents(void)

텍스트이(가) 표시된 사진

자동 생성된 설명

string의 내용을 리턴하는 멤버 함수로, 생성하면서 string의 내용을 넣어주었던 str을 리턴한다.

7) int Str::compare(class Str& a)

텍스트이(가) 표시된 사진

자동 생성된 설명

Class a에 들어간 string의 내용과 str을 비교하는 멤버 함수이다. strcmp을 이용하여 내용이 같으면 0을 반환한다.

8) void Str::operator=(char \*)

텍스트이(가) 표시된 사진

자동 생성된 설명

string의 값을 대입하는 멤버 함수이다. 대입하려는 string이 비어 있는지 strlen으로 판단을 하고, 비어 있다면 에러 메시지를 출력하고 len에 0을 넣고 delete를 이용하여 할당한 str을 할당 해제한다. 비어 있지 않으면, 이전에 할당한 str을 delete로 할당 해제하고, len에 넣으려는 a의 길이를 넣어 준 후, len 만큼 chat형의 메로리를 할당하여 char형 포인터인 str에 넣어준다. str에 a의 내용을 넣어주기 위해 strcpy를 이용하여 문자열을 str에 복사한다.

9) void Str::operator=(class Str& a)

텍스트이(가) 표시된 사진

자동 생성된 설명

Str의 내용을 대입하는 멤버 함수이다. Str의 내용을 리턴하는 멤버 함수인 contents()의 길이를 strlen으로 재어, 비어 있으면 에러 메시지를 출력하고 len에 0을 넣어준 후, 할당했던 str을 delete로 할당 해제한다. 비어 있지 않으면, 이전에 할당한 str을 할당 해제하고, len에 contents()의 함수의 길이를 넣어준 후, chat형 포인터인 str에 len만큼의 chat형 메모리 공간을 할당하여 넣어준다. str에 contents()의 값을 넣어주기 위해 strcpy를 이용하여 문자열을 복사한다.

10) Makefile

텍스트, 모니터, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

컴파일 시 사용할 명령어인 g++과 -c를 CC와 CFLAG 매크로로 정의한다. 실행 파일의 이름을 main으로 지정하기 위해 TARGET 매크로를 정의한다. 오브젝트 파일의 이름을 지정하여 OBJS 매크로로 하나로 묶어준다. TARGET 파일을 생성할 때, OBJS 파일에 의존한다. “$(CC) -o $@ $^” 명령어에 의해 의존 목록 파일 전체에 대응하여 각 .c 파일마다 파일의 이름을 그대로 사용한 .o 파일을 생성하여 실행 파일을 생성한다. “$(CC) $(CFLAG) -o $@ $<” 명령어에 의해 .c 파일을 이용하여 .o 파일을 생성한다. “make clean” 명령어를 실행할 때 오브젝트 파일과 실행 파일을 지워주기 위해 “rm $(OBJS) $(TARGET)” 문장을 사용하였다. 또한 clean이라는 파일에 의해 “make clean” 명령어가 수행이 되지 않는 것을 방지하기 위해 .PHONY를 정의하였다.

11) main 함수로 테스트한 결과는 아래의 사진과 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명

(main 함수 (main.cpp))

텍스트이(가) 표시된 사진

자동 생성된 설명