|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2018\_2\_C++ \_06 | 학번 : | 20157135 | 이름 : | 이상우 |

1. 복사 생성자

|  |
| --- |
| #include <iostream>  using namespace std;  class Circle {  private:  int radius;  public:  Circle(Circle& c); // 복사 생성자 선언  Circle() { radius = 1; }  Circle(int radius) { this->radius = radius; }  double getArea() { return 3.14\*radius\*radius; }  };  Circle::Circle(Circle& c) { // 복사 생성자 구현  this->radius = c.radius;  cout << "복사 생성자 실행 radius = " << radius << endl;  }  int main() {  Circle src(30); // src 객체의 보통 생성자 호출  Circle dest(src); // dest 객체의 복사 생성자 호출  cout << "원본의 면적 = " << src.getArea() << endl;  cout << "사본의 면적 = " << dest.getArea() << endl;  } |
| **[실행결과] - 복사 생성자** |
| **[실행결과] - 디폴트 복사 생성자** |

* 얕은 복사로 문제가 발생하는 예제

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include <iostream>  #include <cstring>  using namespace std;  class Person { // Person 클래스 선언  char\* name;  int id;  public:  Person(int id, const char\* name); // 생성자  ~Person(); // 소멸자  void changeName(const char \*name);  void show() { cout << id << ',' << name << endl; }  };  Person::Person(int id, const char\* name) { // 생성자  this->id = id;  int len = strlen(name); // name의 문자 개수  this->name = new char[len + 1]; // name 문자열 공간 핟당  strcpy(this->name, name); // name에 문자열 복사  }  Person::~Person() {// 소멸자  if (name) // 만일 name에 동적 할당된 배열이 있으면  delete[] name; // 동적 할당 메모리 소멸  }  void Person::changeName(const char\* name) { // 이름 변경  if (strlen(name) > strlen(this->name))  return;  strcpy(this->name, name);  }  int main() {  Person father(1, "Kitae"); // (1) father 객체 생성  Person daughter(father); // (2) daughter 객체 복사 생성. 복사생성자호출  cout << "daughter 객체 생성 직후 ----" << endl;  father.show(); // (3) father 객체 출력  daughter.show(); // (3) daughter 객체 출력  daughter.changeName("Grace"); // (4) daughter의 이름을 "Grace"로 변경  cout << "daughter 이름을 Grace로 변경한 후 ----" << endl;  father.show(); // (5) father 객체 출력  daughter.show(); // (5) daughter 객체 출력  return 0; // (6), (7) daughter, father 객체 소멸  } |

* 얕은 복사의 문제를 해결한 깊은 복사

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include <iostream>  #include <cstring>  using namespace std;  class Person { // Person 클래스 선언  char\* name; int id;  public:  Person(int id, const char\* name); // 생성자  Person(Person& person); // 복사 생성자  ~Person(); // 소멸자  void changeName(const char \*name);  void show() { cout << id << ',' << name << endl; }  };  Person::Person(int id, const char\* name) { // 생성자  this->id = id;  int len = strlen(name); // name의 문자 개수  this->name = new char[len + 1]; // name 문자열 공간 핟당  strcpy(this->name, name); // name에 문자열 복사  }  **Person::Person(Person& person) { // 복사 생성자**  **this->id = person.id; // id 값 복사**  **int len = strlen(person.name);// name의 문자 개수**  **this->name = new char[len + 1]; // name을 위한 공간 핟당**  **strcpy(this->name, person.name); // name의 문자열 복사**  **cout << "복사 생성자 실행. 원본 객체의 이름 " << this->name << endl;**  **}**  Person::~Person() {// 소멸자 코드 생략  }  void Person::changeName(const char\* name) { // 이름 변경  if (strlen(name) > strlen(this->name))  return; // 현재 name에 할당된 메모리보다 긴 이름으로 바꿀 수 없다.  strcpy(this->name, name);  }  int main() {  Person father(1, "Kitae"); // (1) father 객체 생성  Person daughter(father); // (2) daughter 객체 복사 생성. 복사생성자호출  cout << "daughter 객체 생성 직후 ----" << endl;  father.show(); // (3) father 객체 출력  daughter.show(); // (3) daughter 객체 출력  daughter.changeName("Grace"); // (4) daughter의 이름을 "Grace"로 변경  cout << "daughter 이름을 Grace로 변경한 후 ----" << endl;  father.show(); // (5) father 객체 출력  daughter.show(); // (5) daughter 객체 출력  return 0; // (6), (7) daughter, father 객체 소멸  } |

* 내용 점검

1. 교재 5장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 13 | 디폴트 생성자 : MyClass() , 복사 생성자 : MyClass(MyClass&) |
| 14 | 2. void f(MyClass \*p); |
| 15 | 1. ~MyClass(){ delite[] element; } 2. MyClass(MyClass &b){this->size=b.size; element =b. element; } 3. MyClass(MyClass &b){   this->size=b.size;  element =new element[size]  For(int i=0;i<size;i++)  Element[i]=b. element[i]; } |
| 16 | 1.복사 생서자는 중복이 가능하여 필요에 따라 여러 개 선언될수 있다. |
| 17 | 1. Student(Student&b){name=b. name;   id=b.id;  gread = b. gread; } |
| 18 | 1. Student(Student&b){   pname=b. pname;  pid=b.pid;  pgread = b. pgread; |
| 19 | 실행오류 발생 비 정상적으로 종료 된다. |

* 프로그램 과제

1. 교재 274p : 문제11번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <cstring>  #pragma warning (disable:4996) //strcpy 보안 문제로 못쓰게 하는 것을 쓸수있도록 하게함.  ///error C4996: 'strcpy': This function or variable may be unsafe. Consider using strcpy\_s instead.  ///To disable deprecation, use \_CRT\_SECURE\_NO\_WARNINGS. See online help for details.  using namespace std;  class Book {  char \*title; // 제목  int price; // 가격  public:  Book(const char\* t, int p);  Book(Book& b); // 문제 3번 답. 깊은 복사 복사 생성자 작성  ~Book();  void set(const char\* t, int p);  void show() { cout << title << ' ' << price << "원" << endl; }  };  // 1번 문제  Book::Book(const char\*t, int p) {// 문제 1번 답 (Book class 의 생성자)  int length = strlen(t); // t 의 길이를 length 에 저장 (정수형 변수)  this->title = new char[length + 1];//length +1 만큼의 새로운 배열 생성(메모리 할당)  strcpy(this->title, t);//this->title 를 t 에 복사 한다.  this->price = p; // price 에 p 를 저장  }  Book::~Book() { // 문제 1번 답(Book 의 소멸자)  if (title)  delete[] title;  }  void Book::set(const char\* t, int p) { // 문제 1번 답 (set 함수)  if (this->title)  delete[] this->title; // title에 할당된 메모리 반환  int length = strlen(t); // t 의 길이를 length 에 저장 (정수형 변수)  this->title = new char[length + 1]; //length +1 만큼의 새로운 배열 생성(메모리 할당)  strcpy(this->title, t);//this->title 를 t 에 복사 한다.  this->price = p;// price 에 p 를 저장  }  // 1번 문제 end  // 문제 2번 답 (디폴트 복사 생성자)  /\*  Book::Book(Book& book)  title = book.title;  price = book.price;  }  \*/  // 문제 3번 답( 깊은 복사 복사 )  Book::Book(Book& book) {  int length = strlen(book.title); //title 의 길이를 length 에 저장(정수형 변수)  title = new char[length + 1];//length +1 만큼의 새로운 배열 생성(메모리 할당)  strcpy(title, book.title);// book.title 를 title 에 복사 한다.  price = book.price; // price 에 매개변수로 받은 book 객체의 price 저장  }  int main() {  Book cpp("명품C++", 10000);  Book java = cpp;  java.set("명품자바", 12000);  cpp.show();  java.show();  char c = getchar();  }  // 문제 4번 정답  #include <iostream>  #include <string>  using namespace std;  class Book {  string title; // 제목  int price; // 가격  public:  Book(string t, int p);  ~Book();  void set(string t, int p);  void show() {  cout << title << ' ' << price << "원" << endl;  }  };  Book::Book(string t, int p) { // 문제 4번 답 (Book class 생성자)  this->title = t; // 현재 필드(title)에 매개변수로 받은 t 저장  this->price = p; // 현재 필드(price)에 매개변수로 받은 p 저장  }  Book::~Book() {} // 문제 4번 답 (Book class 소멸자)  void Book::set(string t, int p) { // 문제 4번 답 (set 함수)  this->title = t;// 현재 필드(title)에 매개변수로 받은 t 저장  this->price = p;// 현재 필드(price)에 매개변수로 받은 p 저장  }  int main() {  Book cpp("명품C++", 10000);  Book java = cpp;  java.set("명품자바", 12000);  cpp.show();  java.show();  char c = getchar();  } |
| **[실행 결과]** |

1. 교재 275p : 문제12번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  using namespace std;  class Dept {  int size; //scores 배열 크기  int \*scores;//동적 할당 받을 정수 배열의 정수  public:  Dept(int size) {// 생성자  this->size = size;  scores = new int[size];  }  Dept(Dept &dept); //복사 생성자 // 12.2  ~Dept(); // 소멸자  int getSize() {  return size;  }  void read(); // size만큼 키보드에서 정수를 읽어 scores 배열에 저장  bool isOver60(int index); // index의 학생의 성적이 60보다 크면 return  };  void Dept::read() {  cout << "10개 점수 입력 >> ";  for (int i = 0; i < size; i++) { // size개수  cin >> scores[i]; // 성적을 score배열에 입력  }  }  bool Dept::isOver60(int index) {  if (scores[index] > 60)  return true; // 성적이 60 이상이면 true  else  return false; // 성적이 60 미만이면 false  }  Dept::Dept(Dept &dept) { // 복사 생성자 // 12.2  this->size = dept.size; // size 복사  scores = new int[size]; // score를 위한 동적 할당  for (int i = 0; i < size; i++) { // size 만큼 배열 원소 복사  this->scores[i] = dept.scores[i];  }  }  int countPass(Dept dept) { // dept 학과 60점 이상으로 통과하는 학생의 수 리턴  int count = 0;  for (int i = 0; i < dept.getSize(); i++) {// size 만큼 for문을 돌리고 60점 이상일때 count 를 1씩 증가  if (dept.isOver60(i))  count++;  }  return count;  }  Dept::~Dept() { // 소멸자 구현  delete[] scores;  }  int main() {  Dept com(10); // 총 10명이 있는 학과 com  com.read(); // 총 10명의 학생들의 성적을 키보드로부터 읽어 scores 배열에 저장  int n = countPass(com); // com 학과에 60점 이상으로 통과한 학생 수를 return  cout << "60점 이상은 " << n << "명";  char c = getchar();  }  //12-2  /\*이미 반환된 메모리를 다시 반환 하려 하기때문에 에러가 발생 한다.\*/  //12-3  int Dept::countPass() { // dept 학과 60점 이상으로 통과하는 학생의 수 리턴\_countPass(Dept dept)에서  // 매개 변수 Dept dept 를 제거  int count = 0;  for (int i = 0; i < getSize(); i++) {// size 만큼 for문을 돌리고 60점 이상일때 count 를 1씩 증가  if (isOver60(i))  count++;  }  return count;  }  int n = com.countPass(); // countPass() 함수 의 인수 com 을 제거 |
| **[실행 결과]** |