|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_06 | 학번 : | 20145128 | 이름 : | 박인근 |

* **강의 내용**

|  |
| --- |
| 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]** |

|  |
| --- |
| 1. 깊은 복사 |
| #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  } |
| **[실행결과]** |

|  |
| --- |
| 1. 디폴트 매개변수 |
| #include <iostream>  using namespace std;  class MyVector{  int \*p;  int size;  public:  MyVector(int n=100) {  p = new int [n];  size = n;  }  ~MyVector() { delete [] p; }  };  int main() {  MyVector \*v1, \*v2;  v1 = new MyVector(); // 디폴트로 정수 배열 100 동적 할당  v2 = new MyVector(1024); // 정수 배열 1024 동적 할당  delete v1;  delete v2;  } |
| **[실행결과]**    오류 없이 잘 실행이 됩니다.  디폴트 생성자로 v1 에는 정수배열 100을 동적 할당 받고 v2는 정수배열 1024를 동적 할당 한 후 둘 다 메모리 할당 해제를 해줍니다. 따라서 실행결과는 아무것도 나오지 않습니다. |

|  |
| --- |
| 1. static 멤버 사용하기 |
| class Person {  public:  double money; // 개인 소유의 돈  void addMoney(int money) { this->money += money; }  static int sharedMoney; // 공금  static void addShared(int n) {  sharedMoney += n;  }  };  // static 변수 생성. 전역 공간에 생성  int Person::sharedMoney=10; // 10으로 초기화  int main() {  // static 멤버 접근, 공금=60, han 객체가 생기기전부터 static 멤버접근  **Person::addShared(50);**  cout << **Person::sharedMoney** << endl;  Person han;  han.money = 100;  han.sharedMoney = 200; // static 멤버 접근, 공금=200  **Person::sharedMoney = 300;** // static 멤버 접근, 공금=300  **Person::addShared(100);** // static 멤버 접근, 공금=400  cout << han.money << ' ' << **Person::sharedMoney** << endl;  } |
| **[실행결과]** |

* 내용 점검 – 6장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | 2 |
| 2 | 3 |
| 3 | 3,4 |
| 4 | 1,2 |
| 5 | 2 |
| 6 | 2 |
| 7 | 4 |
| 8 | 2 |
| 9 | 3 |
| 10 | 2 |
| 11 | 4 |
| 12 | 1. 0 (2) 3 (3) 5hello |
| 13 | int f(int a, int b=0) 모호하다, 함수 호출시 ex) f(10) int f(int a) 의 매개변수에 10을 넘겨주는 것인지 int f(int a, int b=0) 의 a에 10을 넘겨주고 b는 디폴트 값을 주는 것인지 알 수 없기 때문이다. |
| 14 | 정수형(int) 가 매개변수로 전달될 경우 float 형으로 형 변환 할 것인지 double로 형 변환 할 것인지 알 수 없어 오류가 발생할 것이다. |
| 15 | 4 , getB() 는 static 형이 아닌 int형 이기 때문에 |
| 16 | 3 |

* 프로그램 과제

|  |
| --- |
| 1. 교재 274p 문제11번 |
| **[프로그램 소스]**  3.  #include <iostream>  #include <string>  #include <cstring>  using namespace std;  class Book {  char \*title; // 제목 문자열  int price;  public:  Book(const char\* title, int price);  Book(Book& b);  ~Book();  void set(const char\* title, int price);  void show() { cout << title << ' ' << price << "원" << endl; }  };  Book::Book(const char\* title, int price) {  int length = strlen(title);  this->title = new char[length + 1];  strcpy(this->title, title);  this->price = price;  }  void Book::set(const char\* title, int price) {  if (this->title) {  delete this->title;  }  int length = strlen(title);  this->title = new char[length + 1];  strcpy(this->title, title);  this->price = price;  }  Book::Book(Book& b) { // 복사 생성자  this->price = b.price; // price 값 복사  int length = strlen(b.title); // title의 문자 개수  this->title = new char[length + 1]; // title을 위한 공간 할당  strcpy(this->title, b.title); // title의 문자열 복사  }  Book::~Book() { delete title; }  int main()  {  Book cpp("명품 C++", 10000);  Book java = cpp; // 복사 생성자가 호출됨  java.set("명품자바", 12000);  cpp.show();  java.show();  system("pause");  }  4.  #include <iostream>  #include <string>  #include <cstring>  using namespace std;  class Book {  string title; // 제목 문자열  int price;  public:  Book(string title, int price);  //Book(Book& b);  ~Book();  void set(string title, int price);  void show() { cout << title << ' ' << price << "원" << endl; }  };  Book::Book(string title, int price) {  this->title = title;  this->price = price;  }  void Book::set(string title, int price) {  this->title = title;  this->price = price;  }  Book::~Book() { }  int main()  {  Book cpp("명품 C++", 10000);  Book java = cpp; // 복사 생성자가 호출됨  java.set("명품자바", 12000);  cpp.show();  java.show();  system("pause");  } |
| **[실행결과]**  3. 깊은 복사 생성자 작성시    4. string 클래스 사용 |

|  |
| --- |
| 1. 교재 275p 문제12번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  #include <cstring>  using namespace std;  class Dept {  int size; // scores 배열의 크기  int\* scores; // 동적 할당 받을 정수 배열의 주소  public:  Dept(int size) {  this->size = size;  scores = new int[size];  }  Dept(Dept& dept); //복사 생성자  ~Dept(); //소멸자  int getSize() { return size; }  void read(); // size만큼 키보드에서 정수를 읽어 scores 배열에 저장  bool isOver60(int index); // index의 학생의 성적이 60보다 크면 true 리턴  };  void Dept::read() {  cout << size << "개의 정수 입력>> ";  for (int i = 0; i < size; i++) {  int n;  cin >> n;  scores[i] = n;  }  }  bool Dept::isOver60(int index) {  if (scores[index] > 60) return true;  else  return false;  }  int countPass(Dept dept) { // dept 학과에 60점 이상으로 통과하는 학생의 수 리턴  int count = 0;  for (int i = 0; i < dept.getSize(); i++) {  if (dept.isOver60(i)) count++;  }  return count;  }  Dept::Dept(Dept& dept) {  this->size = dept.size;  this->scores = new int[dept.size];  for (int i = 0; i < dept.size; i++)  this->scores[i] = dept.scores[i];  }  Dept::~Dept() { 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 학생의 수를 리턴  cout << "60점 이상은 " << n << "명" << endl;  system("pause");  } |
| **[실행결과]**         1. int n = countPass(com); 에서 실행될 때 복사생성자가 호출됩니다.   복사 생성자를 지우게 될 경우 디폴트 복사 생성자(얕은 복사) 로 컴파일 해야 하는데 생성자에서 동적 메모리를 할당 받은 멤버 가 있기 때문에 오류가 납니다.   1. 복사 생성자를 지우고 자동으로 생성되는 디폴트 복사 생성자를 이용합니다. |

|  |
| --- |
| 1. 교재 313p 문제2번 |
| **[프로그램 소스]**  (1)  #include <iostream>  #include <string>  #include <cstring>  using namespace std;  class Person {  int id;  double weight;  string name;  public:  void show() {  cout << id << ' ' << weight << ' ' << name << endl;  }  Person() { this->id = 1; this->weight = 20.5; this->name = "Grace"; }  Person(int id, string name) { this->id = id; this->name = name; this->weight = 20.5; }  Person(int id, string name, double weight) {  this->id = id;  this->name = name;  this->weight = weight;  }  };  int main()  {  Person grace, ashley(2, "Ashley"), helen(3, "Helen", 32.5);  grace.show();  ashley.show();  helen.show();  system("pause");  }  (2)  #include <iostream>  #include <string>  #include <cstring>  using namespace std;  class Person {  int id;  double weight;  string name;  public:  void show() {  cout << id << ' ' << weight << ' ' << name << endl;  }  Person(int id = 1, string name = "Grace" , double weight = 20.5) {  this->id = id;  this->name = name;  this->weight = weight;  }  };  int main()  {  Person grace, ashley(2, "Ashley"), helen(3, "Helen", 32.5);  grace.show();  ashley.show();  helen.show();  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 313p 문제3번 |
| **[프로그램 소스]**  (1)  #include <iostream>  #include <string>  #include <cstring>  using namespace std;  int big(int a, int b) {  int c = 100;  if (a > b) {  if (a > c) return c;  else return a;  }  else {  if (b > c) return c;  else return b;  }  }  int big(int a, int b, int c) {  if (a > b) {  if (a > c) return c;  else return a;  }  else {  if (b > c) return c;  else return b;  }  }  int main()  {  int x = big(3, 5); // 3과 5중 큰 값 5는 최대값 100보다 작으므로, 5 리턴  int y = big(300, 60); //300과 60중 큰 값 300이 최대값 100보다 크므로, 100 리턴  int z = big(30, 60, 50); // 30과 60 중 큰 값 60이 최대값 50보다 크므로, 50 리턴  cout << x << ' ' << y << ' ' << z << endl;  system("pause");  }  (2)  #include <iostream>  #include <string>  #include <cstring>  using namespace std;  int big(int a, int b, int c = 100) {  if (a > b) {  if (a > c) return c;  else return a;  }  else {  if (b > c) return c;  else return b;  }  }  int main()  {  int x = big(3, 5); // 3과 5중 큰 값 5는 최대값 100보다 작으므로, 5 리턴  int y = big(300, 60); //300과 60중 큰 값 300이 최대값 100보다 크므로, 100 리턴  int z = big(30, 60, 50); // 30과 60 중 큰 값 60이 최대값 50보다 크므로, 50 리턴  cout << x << ' ' << y << ' ' << z << endl;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 316p 문제6번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  #include <cstring>  using namespace std;  class ArrayUtility2 {  public:  //s1과 s2를 연결한 새로운 배열을 동적 생성하고 포인터 리턴  static int\* concat(int s1[], int s2[], int size) {  int retSize = size\*2;  int\* s3 = new int[retSize];  for (int i = 0; i < size; i++) {  s3[i] = s1[i];  }  for (int i = 0; i < size; i++) {  s3[i + size] = s2[i];  }  return s3;  }  //s1에서 s2에 있는 숫자를 모두 삭제한 새로운 배열을 동적 생성하여 리턴, 리턴하는 배열의 크기는  //retSize에 전달, retSize가 0인 경우 NULL 리턴  static int\* remove(int s1[], int s2[], int size, int& retSize) {  int\* s3 = new int[size];  retSize = size;  for (int i = 0; i < size; i++) {  for (int j = 0; j < size; j++) {  if (s1[i] == s2[j]) {  retSize -= 1;  break;  }  }  s3[i] = s1[i];  }  return s3;  }  };  int main()  {  int x[5], y[5], \*c, \*r, size;  cout << "정수를 5 개 입력하라. 배열 x에 삽입한다>>>";  for (int i = 0; i < 5; i++) {  cin >> x[i];  }  cout << "정수를 5 개 입력하라. 배열 y에 삽입한다>>>";  for (int i = 0; i < 5; i++) {  cin >> y[i];  }  cout << "합친 정수 배열을 출력한다." << endl;  c = ArrayUtility2::concat(x, y, 5);  for (int i = 0; i < 10; i++){  cout << c[i] << ' ';  }  cout << endl;  cout << "배열 x[]에서 y[]를 뺀 결과를 출력한다. 개수는 2" << endl;  r = ArrayUtility2::remove(x, y, 5, size);  for (int i = 0; i < size; i++) {  cout << r[i] << ' ';  }  cout << endl;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 316p 문제7번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  #include <time.h>  using namespace std;  class Random {  public:  //항상 다른 랜덤수를 발생시키기 위한 seed를 설정하는 함수  static void seed() { srand((unsigned)time(0)); } //씨드 설정  static int nextInt(int min = 0, int max = 32767) { //min과 max 사이의 랜덤 정수 리턴  int a = rand() % (max - min + 1) + min;  return a;  }  static char nextAlphabet() { // 랜덤 알파벳 문자 리턴  char a = rand() % 2; // 대, 소문자  if (a == 0) { //대문자  a = rand() % 26 + 65; //아스키코드 65~90 대문자 알파벳  return a;  }  else { //소문자  a = rand() % 26 + 97; //아스키코드 97~122 소문자 알파벳  return a;  }    }  static double nextDouble() { // 0보다 크거나 같고 1보다 적은 랜덤 실수 리턴  double a = rand() / (double)RAND\_MAX \* 1;  return a;  }  };  int main()  {  int min, max, a, size = 10;  char b;  double c;  Random::seed(); // 랜덤한 수 계속 생성  cout << "1에서 100까지 랜덤한 정수 10개를 출력합니다." << endl;    for (int i = 0; i < size; i++) {  a = Random::nextInt(1, 100);  cout << a << ' ';  }  cout << endl;  cout << "알파벳을 랜덤하게 10개 출력합니다." << endl;  for (int i = 0; i < size; i++) {  b = Random::nextAlphabet();  cout << b << ' ';  }  cout << endl;  cout << "랜덤한 실수를 10개 출력합니다." << endl;  for (int i = 0; i < size; i++) {  c = Random::nextDouble();  if (i == 5) cout << endl;  cout << c << ' ';  }  cout << endl;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 318p 문제9번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Board {  static string \*p;  static int size;  public:  static void add(string s) {  p[size] = s;  size++;  }  static void print() {  cout << "\*\*\*\*\*\*\*\*\*\* 게시판입니다. \*\*\*\*\*\*\*\*\*\*"<< endl;  for (int i = 0; i < size; i++) {  cout << i << ": " << p[i] << endl;  }  }  };  string\* Board::p = new string[10];  int Board::size = 0;  int main()  {  // Board myBoard; // 객체 생성은 컴파일 오류입니다. static 이기 때문입니다.  Board::add("중간고사는 감독 없는 자율 시험입니다.");  Board::add("코딩 라운지 많이 이용해 주세요.");  Board::print();  Board::add("진소린 학생이 경진대회 입상하였습니다. 축하해주세요");  Board::print();  system("pause");  } |
| **[실행결과]** |