|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_06 | 학번 : | 20135151 | 이름 : | 이갑성 |

* **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]**  (없음) |

|  |
| --- |
| 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 | 2번 |
| 2 | 3번 |
| 3 | 3, 4번 |
| 4 | 1, 2번 |
| 5 | 2번 |
| 6 | 2번 |
| 7 | 4번 |
| 8 | 2번 |
| 9 | 3번 |
| 10 | 2번 |
| 11 | 4번 |
| 12 | (1) 0  (2) 3  (3) 5 hello |
| 13 | f(10); |
| 14 | area(10); |
| 15 | 4번 |
| 16 | 3번 (범위 연산자 쓸때는 클래스명으로 접근하여야 한다.) |

* 프로그램 과제

|  |
| --- |
| 1. 교재 274p 문제11번 |
| **[프로그램 소스]**  (2)  Book::Book(Book& book) {  this->title = book.title;  this->price = book.price;  }  (3)  Book::Book(Book& book) {  int len;  len = strlen(book.title);  this->title = new char[len+1]; //책 제목 메모리 할당  strcpy(this->title, book.title); //책 제목 복사해줌  this->price = book.price;  }  (4)  #include <iostream>  #include <cstring>  #include <string>  using namespace std;  class Book {  string title;  int price;  public:  Book(string title, int price);  //~Book();  void set(string title, int price);  void show() {  cout << title << ' ' << price << "원" << endl;  }  //Book(Book& book);  };  Book::Book(string title, int price) { //생성자  this->title = title;  this->price = price;  }  void Book::set(string title, int price) {  this->title = title;  this->price = price;    }  int main() {  Book cpp("명품C++ ", 10000);  Book java(cpp);  java.set("명품자바", 12000);  cpp.show();  java.show();  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 275p 문제12번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  class Dept {  int size;  int\* scores;  public:  Dept(int size) {  this->size = size;  scores = new int[size];  }  ~Dept();  int getSize() {  return size;  }  int\* getScore() {//추가  return scores;  }  void read();  bool isOver60(int index);  };  void Dept::read() {  cout << "10개 점수 입력 >> ";  for (int i = 0; i < 10; i++) {  cin >> scores[i];  }  }  bool Dept::isOver60(int index) {  if (scores[index] > 60) {  return true;  }  else return false;  }  Dept::~Dept() { //소멸자  delete[] scores;  }  int countPass(Dept& dept) {  int count = 0;  for (int i = 0; i < dept.getSize(); i++)  {  if (dept.isOver60(i))  count++;  }  return count;  }  int main() {  Dept com(10);  com.read();  int n = countPass(com);  cout << "60점 이상은 " << n << "명";  } |
| **[실행결과]**  (2) 객체를 매개변수로 보냈을 때 묵시적으로 복사생성자가 호출되기 때문이다.  (3) |

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

|  |
| --- |
| 1. 교재 313p 문제3번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  int big(int a, int b, int max = 100) {  if (a > b) {  if (a < max) //a가 b보다 큰 경우, a가 max보다 작으면 a를 리턴, max보다 크면 max를 리턴  return a;  else  return max;  }  else {  if (b < max)  return b;  else  return max;  }  }  int main() {  int x = big(3, 5);  int y = big(300, 60);  int z = big(30, 60, 50);  cout << x << ' ' << y << ' ' << z << endl;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 316p 문제6번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  class ArrayUtility2 {  public:  static int\* concat(int s1[], int s2[], int size);  static int\* remove(int s1[], int s2[], int size, int& retSize);  };  int\* ArrayUtility2::concat(int s1[], int s2[], int size) {  int \*s3 = new int[size]; //합칠 배열 동적메모리 할당  for (int i = 0; i < 5; i++) {  s3[i] = s1[i];  }  for (int i = 0; i < 5; i++) {  s3[i + 5] = s2[i];  }  return s3;  delete[] s3;  }  int\* ArrayUtility2::remove(int s1[], int s2[], int size, int& retSize) {  bool exist = false; // 동일한 값의 유무를 판단  int \*result = new int[size];  for (int i = 0; i < size; i++) {  for (int j = 0; j < size; j++) {  if (s1[i] == s2[j])  {  exist = true;  }  }  if (exist == false) {  result[retSize] = s1[i];  retSize++;  }  exist = false;  }  return result;  delete[] result;  }  int main() {  int \*arr1 = new int[5];  int \*arr2 = new int[5];  int \*arr3;  int \*arr4 = new int[5];  int cnt = 0;  cout << "정수를 5 개 입력하여라. 배열 x에 삽입한다 >> ";  for (int i = 0; i < 5; i++) {  cin >> arr1[i];  }  cout << "정수를 5 개 입력하여라. 배열 y에 삽입한다 >> ";  for (int i = 0; i < 5; i++) {  cin >> arr2[i];  }  arr3 = ArrayUtility2::concat(arr1, arr2, 10);  cout << "합친 정수 배열을 출력한다." << endl;  for (int i = 0; i < 10; i++) {  cout << arr3[i] << " ";  }  cout << endl;  arr4 = ArrayUtility2::remove(arr1, arr2, 5, cnt);  cout << "배열 x[]에서 y를 뺀 결과를 출력한다. 개수는 " << cnt << endl;  for (int i = 0; i < cnt; i++) {  cout << arr4[i] << " ";  }  delete[] arr1;  delete[] arr2;  delete[] arr3;  delete[] arr4;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 316p 문제7번 |
| **[프로그램 소스]**  #include <iostream>  #include <ctime>  #include <cstdlib>  using namespace std;  class Random {  public:  static void seed() { srand((unsigned)time(0)); }  static int nextInt(int min = 0, int max = 32767); //디폴트 메소드  static char nextAlpahbet();  static double nextDouble();  };  int Random::nextInt(int min, int max) {  int n;  n = rand() % max + min; //난수값 생성  return n;  }  char Random::nextAlpahbet() {  int n;  char ch;  n = rand() % 2; //(0 or 1 난수 출력)  if (n == 0) { //대문자출력  ch = rand() % 26 + 65;  }  else { //소문자출력  ch = rand() % 26 + 97;  }  return ch;  }  double Random::nextDouble() {  double n;  n = (double)rand() / 32767;  return n;  }  int main() {  Random::seed(); //씨드 설정  int \*arr1 = new int[10];  char \*arr2 = new char[10];  double \*arr3 = new double[10];    for (int i = 0; i < 10; i++)  {  arr1[i] = Random::nextInt(1,100);  }  cout << "1에서 100까지 랜덤한 정수 10개를 출력합니다. " << endl;  for (int i = 0; i < 10; i++)  {  cout << arr1[i] << " ";  }    cout << endl;  for (int i = 0; i < 10; i++)  {  arr2[i] = Random::nextAlpahbet();  }  cout << "알파벳을 랜덤하게 10개 출력합니다. " << endl;  for (int i = 0; i < 10; i++)  {  cout << arr2[i] << " ";  }  cout << endl;  for (int i = 0; i < 10; i++)  {  arr3[i] = Random::nextDouble();  }  cout << "랜덤한 실수를 10개를 출력합니다. " << endl;  for (int i = 0; i < 10; i++)  {  if (i == 5)  cout << endl;  cout << arr3[i] << " ";  }    delete[] arr1;  delete[] arr2;  delete[] arr3;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 318p 문제9번 |
| **[프로그램 소스]**  #include <iostream >  #include <string>  using namespace std;  class Board {  public:  static string text[10]; //내용을 담을 정적 string 배열 선언  static int index;  static void add(string str);  static void print();  };  int Board::index = 0; //정적변수 index를 0으로 초기화  string Board::text[10] = { " " }; //정적변수 text 배열을 " "로 모두 초기화  void Board::add(string str) {  Board::text[index] = str;  Board::index++;  }  void Board::print() {  cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 게시판입니다. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;  for (int i = 0; i < Board::index; i++) {  cout << i << ": " << Board::text[i] << endl;  }  cout << endl;  }  int main() {  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();  } |
| **[실행결과]** |