|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_04 | 학번 : |  | 이름 : |  |

* **강의 내용**

|  |
| --- |
| 1. 객체 배열의 동적 생성과 반환 |
| #include <iostream>  using namespace std;  class Circle {  int radius;  public:  Circle();  Circle(int r);  ~Circle();  void setRadius(int r) { radius = r; }  double getArea();  };  double Circle::getArea() {  return 3.14\*radius\*radius;  }  Circle::Circle() {  radius=1;  cout<<”생성자 실행 radius = “<<radius<<endl;  }  Circle::Circle(int r) {  radius=r;  cout<<”생성자 실행 radius = “<<radius<<endl;  }  Circle::~Circle() {  radius=1;  cout<<”소멸자 실행 radius = “<<radius<<두이;  }  int main() {  Circle \*circleArray = new Circle[3]; // Circle 객체 배열 동적 생성, default 생성자 호출  circleArray[0].setRadius(10); circleArray[1].setRadius(20);  circleArray[2].setRadius(30);  for(int i=0; i<3; i++) // 배열의 각 원소 객체의 멤버 접근  cout << "Circle " << i << "의 면적은 " << circleArray[i].getArea() << endl;  Circle \*p= circleArray; // 포인터 p에 배열의 주소값 설정  for(int i=0; i<3; i++) { // 객체 포인터로 배열 접근  cout << "Circle " << i << "의 면적은 " << p->getArea() << endl;  p++; }  delete[] circleArray; //객체 배열 반환  } |
| **[실행결과]** |

|  |
| --- |
| 1. 객체 멤버 동적생성하기 |
| #include <iostream>  using namespace std;  class Dog {  string \*name;  int \*age;  public:  Dog(string n, int a){ //멤버 변수 동적 메모리 할당  name = new string(n); //name = new string{n};  age = new int(a); //age=new int{a}  }  ~Dog() {  delete name; //동적 할당 된 메모리 해제  delete age;  }  int getAge() { return \*age; }  void setAge(int a) { \*age = a; }  };  int main() {  Dog \*p=new Dog("강아지", 2);  cout << "강아지 나이 : " << p->getAge() << endl;  p->setAge(5);  cout << "강아지 나이 : " << p->getAge() << endl;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 문자열 입력 받고 회전시키기 |
| #include <iostream>  #include <string>  using namespace std;  int main() {  string s;  cout << "문자열을 입력하세요(한글 안됨) " << endl;  **getline(cin, s, '\n'); // 문자열 입력**  int len = **s.length()**; **// 문자열의 길이**  for(int i=0; i<len; i++) {  string first = **s.substr(0,1)**; **// 맨 앞의 문자 1개를 문자열로 분리**  string sub = **s.substr(1, len-1)**; **// 나머지 문자들을 문자열로 분리**  s = **sub + first**; **// 두 문자열을 연결하여 새로운 문자열로 만듦**  cout << s << endl;  }  } |
| **[실행결과]** |

|  |
| --- |
| 1. String 객체의 동적 생성 |
| #include <iostream>  #include <string>  using namespace std;  int main() {  string \*p = new string("C++"); // 스트링 객체 동적 생성  string \*names = new string[5]; //객체 배열 동적 생성  cout <<\*p<<endl; // "C++" 출력  p->append(" Great!!"); // p가 가리키는 스트링이 "C++ Great!!"이 됨  cout << \*p<<endl; // "C++ Great!!" 출력  for (int i = 0; i < 5; i++) {  cout << "이름 >> ";  getline(cin, names[i], '\n');  }  string latter = names[0];  for (int i = 1; i < 5; i++) {  if (latter < names[i]) { // 사전 순으로 latter 문자열이 앞에 온다면  latter = names[i]; // latter 문자열 변경  }  }  cout << "사전에서 가장 뒤에 나오는 문자열은 " << latter << endl;  delete p;  delete[] names;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 문자열 find 및 replace |
| #include <iostream>  #include <string>  using namespace std;  int main() {  string s;  cout << "여러 줄의 문자열을 입력하세요. 입력의 끝은 &문자입니다." << endl;  getline(cin, s, '&'); // 문자열 입력  cin.ignore(); //& 뒤에 따라오는 <Enter>키를 제거하기 위한 코드  string f, r;  cout << endl << "find: ";  getline(cin, f, '\n'); // 검색할 문자열 입력  cout << "replace: ";  getline(cin, r, '\n'); // 대치할 문자열 입력  int startIndex = 0;  while (true) {  int fIndex = s.find(f, startIndex); // startIndex부터 문자열 f 검색  if (fIndex == -1)  break; // 문자열 s의 끝까지 변경하였음  // fIndex부터 문자열 f의 길이만큼 문자열 r로 변경  s.replace(fIndex, f.length(), r);  startIndex = fIndex + r.length();  }  cout << s << endl;  } |
| **[실행결과]** |

* 내용 점검 – 4장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | (1)Rect \*p;  (2)p = &r  (3) cout << p->getWidth(); << p-> getHeight(); |
| 2 | (1) q = new Rect(w, h);  (2) cout << q->getArea();  (3) delete q; |
| 3 | 1 |
| 4 | 기본 생성자를 생성합니다. |
| 5 | int sum = 0;  for (int i = 0; i < sizeof(r)/sizeof(Rect); i++) {  sum += r[i].getArea();  } |
| 6 | 4 |
| 7 | 4 |
| 8 | 기본생성자  기본생성자  기본생성자  소멸자  소멸자  소멸자 |
| 9 | 1 |
| 10 | Delete[] p; |
| 11 | 3 |
| 12 | 3 |
| 13 | Location() { this->h = this->w = 0; };  Location(int w, int h) {  this->w = w;  this->h = h;  } |
| 14 | 메모리를 동적 할당 후 해제하지 않아서 시스템의 메모리를 고갈 시킬 때 발생한다. |
| 15 | 1. Delete[] p; 2. Delete p;   for (int i = 0; i < 5; i++) {  p = new int;  cin >> \*p;  delete p;  if (\*p % 2 == 1) break;  } |
| 16 | 1 |
| 17 | Stoi(s1);  Stoi(s2); |
| 18 | 3 |
| 19 | #include <iostream>  #include <string>  using namespace std;  int main() {  string a("My name is Jane.");  char ch = a[2];  cout << ch << endl;  if (a == "My name is Jane."){  cout << "same"<<endl;  }  a += "~~";  a[1] = 'Y';  cout << a;  system("pause");  } |

* 프로그램 과제

|  |
| --- |
| 1. 교재 211p 문제2번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  int main() {  int \*number, i;  double sum = 0;  number = new int[5];  cout << "정수 5개 입력 >>";  for (i = 0; i < 5; i++) {  cin >> number[i];  }  for (i = 0; i < 5; i++) {  sum += number[i];  }  cout << "평균" << sum / 5;  delete[] number;  system("pause");  }} |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 211p 문제4번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Sample {  int \*p;  int size;  public:  Sample(int n) {  size = n; p = new int[n];  }  void read();  void write();  int big();  ~Sample();  };  void Sample::read() {  cout << "값을 입력하세요: ";  for (int i = 0; i < size; i++) {  cin >> p[i];  }  }  void Sample::write() {  cout << endl << "출력합니다: ";  for (int i = 0; i < size; i++) {  cout << p[i]<<" ";  }  }  int Sample::big() {  int big = 0;  for (int i = 0; i < size; i++) {  if (big < p[i]) {  big = p[i];  }  }  return big;  }  Sample::~Sample() {  delete[] p;  }  int main() {  Sample s(10);  s.read();  s.write();  cout << "가장 큰 수는" << s.big() << endl;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 213p 문제9번 |
| **[프로그램 소스]**  #include<iostream>  #include<string>  using namespace std;  class Person {  string name;  string tel;  public:  //Person();  string getName() { return name; }  string getTel() { return tel; }  void set(string name, string tel) { this->name = name; this->tel = tel; }  };  int main() {  Person per[3];  int n;  string name, phone;  cout << "이름과 전화 번호를 입력해 주세요" << endl;  for (int i = 0; i < 3; i++) {  cout << "사람 " << i + 1 << ">> ";  cin >> name >> phone;  per[i].set(name, phone);  }  cout << "모든 사람의 이름은 ";  for (int i = 0; i < 3; i++)  cout << per[i].getName() << " ";  cout << endl << "전화번호를 검색합니다. 이름을 입력하세요>>";  cin >> name;  for (int i = 0; i < 3; i++)  if (per[i].getName() == name) n = i;  cout << "전화 번호는 " << per[n].getTel() << endl;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 214p 문제10번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Person {  string name;  public:  Person() {}  Person(string name) { this->name = name; }  string getName() { return name; }  void setName(string name) { this->name = name; }  };  class Family {  Person \*p;  int size;  string name;  public:  Family(string name, int size) {  p = new Person[size];  this->size = size;  this->name = name;  }  void setName(int n, string name) { p[n].setName(name); }  void show();  ~Family() { delete[] p; }  };  void Family::show() {  cout << name << "가족은 다음과 같이 " << size << "명 입니다." << endl;  for (int i = 0; i < size; i++)  cout << p[i].getName() << "\t";  cout << endl;  }  int main() {  Family \*simpson = new Family("Simpson", 3);  simpson->setName(0, "Mr. Simpson");  simpson->setName(1, "Mrs. Simpson");  simpson->setName(2, "Bart Simpson");  simpson->show();  delete simpson;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 216p 문제12번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Circle {  int radius;  string name;  public:  void setCircle(string name, int radius) { this->name = name; this->radius = radius; }  double getArea() { return radius \* radius \* 3.14; }  string getName() { return name; }  };  class CircleManager {  Circle \*p;  int size;  public:  CircleManager() {}  CircleManager(int size);  ~CircleManager() { delete[] p; }  void searchByName();  void searchByArea();  };  CircleManager::CircleManager(int size) {  string n, r;  p = new Circle[size];  this->size = size;  for (int i = 0; i < size; i++) {  cout << "원 " << i + 1 << "의 이름과 반지름 >> ";  cin >> n >> r;  p[i].setCircle(n, stoi(r));  }  }  void CircleManager::searchByName() {  string str; int n=0;  cout << "검색하고자 하는 원의 이름 >> ";  cin >> str;  for (int i = 0; i < size; i++)  if (p[i].getName() == str) n = i;  cout << p[n].getName() << "의 면적은 " << p[n].getArea() << endl;  }  void CircleManager::searchByArea() {  int n;  cout << "최소 면적을 정수로 입력하세요 >> ";  cin >> n;  cout << n << "보다 큰 원을 검색합니다." << endl;  for (int i = 0; i < size; i++) {  if (p[i].getArea() > n)  cout << p[i].getName() << "의 면적은 " << p[i].getArea() << ",";  }  cout << endl;  }  int main() {  int n;  cout << "원의 개수 >> ";  cin >> n;  CircleManager cm(n);  cm.searchByName();  cm.searchByArea();  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 218p 문제14번 : 선수 인원수는 프로그램 실행 시 입력 받도록 한다. |
| **[프로그램 소스]**  #include <iostream>  #include <string>  #include <ctime>  #include <cstdlib>  using namespace std;  class Player {  string name[3];  int i;  public:  Player() { i = 0; }  void setName(string namae) { name[i] = namae; i++; }  string getName() {  i++;  if (i >= 3) {  i = 0;  return name[i];  }  else  return name[i];  }  };  class GamblingGame {  int num[50];  public:  GamblingGame() { srand((unsigned)time(0)); }  void Game(int person);  };  void GamblingGame::Game(int person) {  string named;  Player p; string name;  int i = 0;  for (i = 1; i <= person; i++) {  cout << i << " 번째 선수 이름>>";  cin >> name;  p.setName(name);  }  while (1) {  named = p.getName();  cout << named << ":<Enter>" << endl;  getline(cin, name);  for (int i = 0; i < 3; i++)  num[i] = rand() % 3;  cout << "\t\t" << num[0] << "\t" << num[1] << "\t" << num[2] << "\t";  if (num[0] == num[1] && num[1] == num[2]) {  cout << named << "님 승리!!" << endl;  break;  }  else  cout << "아쉽군요!" << endl;  }  }  int main() {  int person;  cout << "\*\*\*\*\* 갬블링 게임을 시작합니다. \*\*\*\*\*" << endl;  cout << "\*\*\*\*\* 인원수를 입력하세요 \*\*\*\*\*" << endl;  cin >> person;  GamblingGame g;  g.Game(person);  system("pause");  } |
| **[실행결과]** |