|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2018\_2\_C++ \_02 | 학번 : | 20157135 | 이상우 |  |

* **실행결과 확인**

1. 클래스 선언 & 구현 & 생성자

|  |
| --- |
| #include <iostream>  using namespace std;  class Rectangle {  public:  int width, height;  Rectangle();  Rectangle(int w, int h);  Rectangle(int length);  bool isSquare();  **}; //반드시 ;을 해야 함**  Rectangle::Rectangle() {  **// Rectangle::Rectangle() : Rectangle(1) {}로 해도 됨**  width = height = 1;  }  Rectangle::Rectangle(int w, int h) {  **//Rectangle::Rectangle(int w, int h):width(w), height(h){ }로 해도 됨**  width = w; height = h;  }  Rectangle::Rectangle(int length) {  width = height = length;  }  bool Rectangle::isSquare() { // 정사각형이면 true를 리턴하는 멤버 함수  if(width == height) return true;  else return false;  }  int main() {  Rectangle rect1;  Rectangle rect2(3, 5);  Rectangle rect3(3);  if(rect1.isSquare()) cout << "rect1은 정사각형이다." << endl;  if(rect2.isSquare()) cout << "rect2는 정사각형이다." << endl;  if(rect3.isSquare()) cout << "rect3는 정사각형이다." << endl;  } |
| **[실행결과]** |

1. namespace, 문자열

|  |
| --- |
| #include <iostream>  #include <string>  #include <cstring>  using namespace std;  namespace A {  void input() {  char name1[20]; //c-스트링 사용  cout << "namespace A ====" << endl;  cout << "비교하고 싶은 문자열을 입력하세요 : ";  cin.getline(name1, 20);  if(strcmp(name1, "namespace")== 0)  cout << "동일한 문자열 "<<endl;  else  cout << "다른 문자열 " << endl;  }  }  namespace B {  void input() {  string name1, name2; //string 객체 사용  cout << "namespace B ====" << endl;  cout << "비교하고 싶은 문자열을 입력하세요 : ";  getline(cin, name1);  if (name1 == "name space")  cout << "동일한 문자열" << endl;  else  cout << "다른 문자열 " << endl;  }  }  int main() {  A::input();  B::input();  return 0;  } |
| **[실행결과]** |

3) 소멸자

* 객체가 소멸될 때 실행
* 반환값과 매개변수 없음
* 객체 생성 순서와 반대로 소멸

|  |
| --- |
| #include <iostream>  using namespace std;  class Circle {  public:  int radius;  Circle();  Circle(int r);  ~Circle(); //소멸자 선언  double getArea();  };  Circle::Circle() {  radius = 1;  cout << "반지름 " << radius << " 원 생성" << endl;  }  Circle::Circle(int r) {  radius = r;  cout << "반지름 " << radius << " 원 생성" << endl;  }  Circle::~Circle() { //소멸자 구현  cout << "반지름 " << radius << " 원 소멸" << endl;  }  double Circle::getArea() {  return 3.14\*radius\*radius;  }  Circle globalDonut(1000); //전역객체  Circle globalPizza(2000);  void f() {  Circle fDonut(100); //지역객체  Circle fPizza(200);  }  int main() {  Circle mainDonut; //지역객체  Circle mainPizza(30);  f();  } |
| **[실행결과]** |

* **내용 점검**

1. 교재 2장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | void main() {} |
| 3 | Return 0; |
| 6 | (1) using std::cout;  (2) using namespace std; |
| 7 | cin>>name; |
| 8 | Cout<< n+200; |
| 10 | (1) 0 (2) 0 (3) 0 (4) x (5) x |
| 11 | #include <cstring> |
| 12 | (1)kitae  (2)kitae |
| 15 | namespace |
| 16 | std |
| 17 | using namespace std; |
| 18 | #include <iostream> |
| 19 | cout<< age<<pDept; |
| 20 | void main() {  for (int i = 0; i < 5; i++) {  for (int j = 0; j < i; j++) {  cout << "\*";  }  cout << "\n";  }  char ch = getchar();  } |

1. 교재 3장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 5 | class Circle {  int rad;  double getarea();  }; // ; 을 추가 한다. |
| 6 | 디폴트 생성자 추가 |
| 7 | Building() {} |
| 8 | calender::calender(){  year = 10;  }  int calender::getyear() {  return year;  } |
| 11 | (1) house::house(int a, int b) {  sine = a;  num = b;  cout << sine << num;  }  (2) house::~house() {  cout << sine << num;  }  (3) b -> c -> a -> d -> ~house(); |
| 12 | C -> B -> A |
| 13 | 디폴트 접근 제어자 는 private 이기 때문에 TV() 에 접근이 불가능 하다.  public :  TV() {  channels = 256;  }  int channels;  TV(int a) {  channels = a;  }  } 식으로 수정 한다. |
| 14 | 디폴트 접근 제어자 는 private 이기 때문에 TV() 에 접근이 불가능 하다.  public:  int channels ;  int color;  TV() {  channels = 256;  }  TV(int a,int b) {  channels = a;  color = b;  }  }; // 로 수정 한다. |

* 프로그램 과제

1. 교재 92p 문제 13번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  using namespace std;  int main() {  int a, b;  cout << "\*\*\*\*\*승리장에 오신 것을 환영합니다\*\*\*\*\* \n";  while(1) {  cout << "짬뽕 :1, 짜장 :2, 군만두 :3, 종료 :4 >>";  cin >> a;  if (a > 4) {  cout << "다시 주문하세요!!\n";  continue;  }  else if (a < 4) {  cout << "몇인분?";  cin >> b;  }  switch (a) {  case 1:  cout << "짬뽕 "<< b << "인분 나왔습니다.\n";  break;  case 2:  cout << "짜장 " << b << "인분 나왔습니다.\n";  break;  case 3:  cout << "군만두 " << b << "인분 나왔습니다.\n";  break;  case 4:  cout << "오늘 영업은 끝났습니다.\n";  char c = getchar();  c = getchar();  return 0;  break;  }  }  } |
| **[실행 결과]** |

1. 교재 92p 문제 14번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  using namespace std;  int main() {  int j, total=0;  char coffee[100];  cout << "에스프레소 2000원, 아메리카노 2300원, 카푸치노 2500원 입니다.\n";  while (1) {  cout << "주문>> ";  cin >> coffee >> j;  //cout << strcmp(coffee, "에스프레소") ;  if (strcmp(coffee, "에스프레소") == 0) {  total += 2000 \* j;  cout << 2000\*j <<"원입니다. 맛있게 드세요\n";  }  else if (strcmp(coffee, "아메리카노") == 0) {  total += 2300 \* j;  cout << 2300 \* j << "원입니다. 맛있게 드세요\n";  }  else if (strcmp(coffee, "카푸치노") == 0) {  total += 2500 \* j;  cout << 2500 \* j << "원입니다. 맛있게 드세요\n";  }  else {  cout << " 없는 매뉴 입니다. 다시 골라 주세요 !";  continue;  }  if (total > 20000) {  cout <<"오늘 " << total<<"원을 판매하여 카페를 닫습니다. 내일봐요~~";  char c = getchar();  c = getchar();  return 0;  }  }  } |
| **[실행 결과]** |

1. 교재 150p 문제 3번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  #include<string>  using namespace std;  class Account {  string name;  int id, balance;//balance : 잔액  public :  Account(string n, int i, int b) {  name = n;  id = i;  balance = b;  }  string getOwner();  int withdraw(int a);  void deposit(int a);  int inquiry();  };  string Account::getOwner() { // 계좌 의 주인 이름  return name;  }  int Account::withdraw(int a) { // 출금  return (balance -= a);  }  void Account::deposit(int a) { // 저금  balance += a;  }  int Account::inquiry() { //  return balance;  }  int main(){  Account a("kitae" ,1,5000);  a.deposit(50000);  cout << a.getOwner() << "의 잔액은" << a.inquiry() << endl;  int money = a.withdraw(20000);  cout << a.getOwner() << "의 잔액은" << a.inquiry() << endl;  char c = getchar();  return 0;  } |
| **[실행 결과]** |

1. 교재 153p 문제 9번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  using namespace std;  class Oval {  int width, height; // width, height 변수 맴버  public :  Oval(int w, int h) { // 너비와 높이 값을 매개변수로 받는 생성자  width = w;  height = h;  }  Oval() { // 너비와 높이 를 1로 초기화 하는 매개변수 없는 생성자  width = 1;  height = 1;  }  ~Oval() {  cout <<"~Oval 소멸자"<< " 너비" << width << endl;  cout << "~Oval 소멸자" << "높이" << height << endl;  }  int getwidth();  int getheight();  void set(int w, int h);  void show();  };  int Oval::getwidth() {  return width;  }  int Oval::getheight() {  return height;  }  //Oval::~Oval() {  //cout <<" 너비" <<width << endl;  //cout << "높이" << height << endl;  //}  void Oval::show() {  cout << "width : " << width << " " << "height : " << height << endl;  }  void Oval::set(int w, int h) {  width = w;  height = h;  }  int main() {  Oval a; //Oval 형 객체 a 생성자 매개변수 없는 것이 호출되어 1,1 로 width와 height 초기화  Oval b(10, 20); //Oval 형 객체 b 10, 20 로 width와 height 초기화  a.set(11, 22); //a의 height와 width 를 11, 22 으로 셋  a.show(); //출력  cout << b.getwidth() << "," << b.getheight() << endl;  return 0;  } |
| **[실행 결과]** |