|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_03 | 학번 : | 20135151 | 이름 : | 이갑성 |

* **강의 내용**

|  |
| --- |
| 1. 위임생성자와 타겟 생성자 & 멤버변수 초기화 |
| #include <iostream>  using namespace std;  class Point {  int x, y;  public:  Point();  Point(int a, int b);  void show() { cout << "(" << x << ", " << y << ")" << endl; }  };  **Point::Point() : Point(0, 0) { }** // 위임 생성자  **Point::Point(int a, int b) : x(a), y(b) { }** // 타겟 생성자    int main() {  Point origin;  Point target(10, 20);  origin.show();  target.show();  } |

|  |
| --- |
| [실행결과] |

|  |
| --- |
| 1. C++ 구조체 - **클래스와 동일. 단, 접근 지정자 생략 시 public이 디폴트 <- (시험 문제 예상)** |
| #include <iostream>  using namespace std;  struct StructCircle { // C++ 구조체 선언  private:  int radius;  public:  StructCircle(int r) { radius = r; } // 구조체의 생성자  double getArea();  };  double StructCircle::getArea() {  return 3.14\*radius\*radius;  }  int main() {  StructCircle waffle(3); **//구조체 변수 선언 시 struct 키워드 생략**  cout << "면적은 " << waffle.getArea();  } |
| **[실행결과]** |

|  |
| --- |
| 1. 조건부 컴파일로 헤더 파일 중복 문제 해결 – Main.cpp를 완성하시오. |
| **//Adder.h**  **#ifndef ADDER\_H**  **#define ADDER\_H**  **class Adder** { // 덧셈 모듈 클래스  int op1, op2;  public:  Adder(int a, int b);  int process();  };  **#endif**  **//Adder.cpp**  #include “Adder.h”  Adder::Adder(int a, int b) {  op1 = a; op2 = b;  }  int Adder::process() {  return op1 + op2;  }  **//Main.cpp – 완성된 소스를 아래에 추가하시오.**  #include <iostream>  using namespace std;  #include "Adder.h"  int main()  {  int num1, num2;  cout << "두 개의 수를 입력하세요 >> ";  cin >> num1 >> num2;  Adder add(num1, num2); //객체 선언  cout << add.process();    return 0;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 객체 포인터 |
| #include <iostream>  using namespace std;  **class Circle** {  int radius;  public:  Circle() { radius = 1; }  Circle(int r) { radius = r; }  double getArea();  };  double Circle::getArea() {  return 3.14\*radius\*radius;  }  int main() {  Circle donut;  Circle pizza(30);  // 객체 이름으로 멤버 접근  cout << donut.getArea() << endl;  // 객체 포인터로 멤버 접근 : -> 사용  **Circle \*p;**  **p = &donut;**  cout << p->getArea() << endl; // donut의 getArea() 호출  cout << (\*p).getArea() <<endl; // donut의 getArea() 호출  **p = &pizza;**  cout << p->getArea() << endl; // pizza의 getArea() 호출  cout << (\*p).getArea() << endl; // pizza의 getArea() 호출  } |
| **[실행결과]** |

|  |
| --- |
| 1. 객체 배열 |
| #include <iostream>  using namespace std;  class Circle {  int radius;  public:  Circle() { radius = 1; }  Circle(int r) { radius = r; }  void setRadius(int r) { radius = r; }  double getArea();  };  double Circle::getArea() {  return 3.14\*radius\*radius;  }  int main() {  Circle circleArray[3]; // (1) Circle 객체 배열 생성, default 생성자 호출  // 배열의 각 원소 객체의 멤버 접근 (2)  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(2)  cout << "Circle " << i << "의 면적은 " << circleArray[i].getArea() << endl;  Circle \*p; // 객체 배열을 가리키는 포인터 선언 (3)  p = circleArray; // 초기화 (4)  for(int i=0; i<3; i++) { // 객체 포인터로 배열 접근 (5)  cout << "Circle " << i << "의 면적은 " << p->getArea() << endl;  p++; }  } |
| **[실행결과]** |

|  |
| --- |
| 1. 동적 메모리 할당 |
| #include <iostream>  using namespace std;  int main() {  cout << "입력할 정수의 개수는?";  int n;  cin >> n; **// 정수의 개수 입력**  if(n <= 0)  return 0;  **int \*p = new int[n]; // n 개의 정수 배열 동적 할당**    if(!p) {  cout << "메모리를 할당할 수 없습니다.";  return 0;  }  for(int i=0; i<n; i++) {  cout << i+1 << "번째 정수: "; **// 프롬프트 출력**  **cin >> p[i]; } // 키보드로부터 정수 입력**    int sum = 0;  for(int i=0; i<n; i++)  **sum += p[i];**  cout << "평균 = " << sum/n << endl;  **delete [] p; // 배열 메모리 반환**  } |
| **[실행결과]** |

* 내용 점검 – 3장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 2 | 3번 |
| 4 | class Circle {  int radius;  int age;  public:  double getArea();  void older();  };  void Circle::older() {  age++;  } |
| 5 | class Circle {  int radius;  double getArea();  }; |
| 6 | class Tower {  int height = 20;  public:  Tower() {  height = 10;  }  }; |
| 7 | class Building {  private:  int floor;  public:  Building() {}  Building(int s) { floor = s; }  };  int mian()  {  Building twin, star;  Building BlueHouse(5), JangMi(14);  } |
| 8 | class Calendar {  private:  int year;  public:  Calendar();  int getYear();  };  Calendar::Calendar() { year = 10; }  int Calendar::getYear(){  return year;  } |
| 9 | 2번 |
| 10 | 3번 |
| 11 | (1)  House::House(int n, int s) {  numOfRooms = n;  size = s;  cout << "numOfRooms의 값: " << numOfRooms << " size의 값: " << size << endl;  }  (2)  House::~House(){  cout << "소멸자 발생 => numOfRooms의 값 : " << numOfRooms << " size의 값 : " << size << endl;  }  (3)  생성: b,c,a,d 소멸: a,d,c,b |
| 12 | 생성:c,b,a 소멸: a,b,c |
| 13 | 디폴트 생성자의 위치가 private으로 선언되어있기에 오류발생 public으로 선언이 이루어져야한다.  #include <iostream>  using namespace std;  class Tv {  public:  int channels;  Tv() { channels = 256; }  Tv(int a) { channels = a; }  };  int main()  {  Tv LG;  LG.channels = 200;  Tv Samsung(100);  } |
| 14 | 변수 channels이 private로 지정이됨으로 접근할 수가 없게 됨으로 public으로 지정해주어 접근한다.  #include <iostream>  using namespace std;  class Tv {  public:  int channels;  int colors;  Tv() { channels = 256; }  Tv(int a, int b) { channels = a; colors = b; }  };  int main()  {  Tv LG;  LG.channels = 200;  LG.colors = 60000;  Tv Samsung(100, 50000);  } |
| 15 | Tv() { channels = 256; }  Tv(int a) { channels = a; } |
| 16 | 2번 |
| 17 | 1번 |
| 18 | 1번 |
| 19 | 4번 |
| 20 | class Family {  int count;  char address[20];  public:  Family();  private:  char tel[11];  }; |

* 프로그램 과제

|  |
| --- |
| 1. 교재 150p 문제2번   힌트) 문자열 길이를 반환하는 인스턴스 메소드 : lenth() or cstring을 이용가능 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Date {  int year, month, day;  string num;  public:  Date() { }; // 디폴트 생성자  Date(int y, int m, int d) {  year = y; month = m; day = d;  }  Date(string str) {  num = str;  }  void show();  int getYear() { return year; }  int getMonth() { return month; }  int getDay() { return day; }  };  void Date::show() {  string y = num.substr(0, 4);  cout << y << "년";  string m = num.substr(5, 1);  cout << m << "월";  string d = num.substr(7, 2);  cout << d << "일" << endl;    }  int main()  {  Date birth(2014, 3, 20);  Date independenceDay("1945/8/15");  independenceDay.show();  cout << endl;  cout << birth.getYear() << ',' << birth.getMonth() << ',' << birth.getDay() << endl;  return 0;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 151p 문제4번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  class CoffeeMachine {  public:  int coffee, water, sugar; //커피, 물, 설탕 선언  CoffeeMachine() { };  CoffeeMachine(int c, int w, int s)  {  coffee = c; water = w; sugar = s;  }  void drinkEspresso();  void drinkAmericano();  void drinkSugarCoffee();  void show();  void fill();  };  void CoffeeMachine::drinkEspresso() {  coffee--;  water--;  }  void CoffeeMachine::drinkAmericano() {  coffee--;  water = water - 2;  }  void CoffeeMachine::drinkSugarCoffee() {  coffee--;  water = water - 2;  sugar--;  }  void CoffeeMachine::fill() {  coffee = 10;  water = 10;  sugar = 10;  }  void CoffeeMachine::show() {  cout << "커피 머신 상태, 커피:" << coffee << " 물:" << water << " 설탕:" << sugar << endl;  }  int main()  {  CoffeeMachine java(5, 10, 3);  java.drinkEspresso();  java.show();  java.drinkAmericano();  java.show();  java.drinkSugarCoffee();  java.show();  java.fill();  java.show();  return 0;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 152p 문제7번   힌트) 디폴트 생성자에서 time() 함수를 사용하여 시드값 설정 – 매번 다른 난수를 생성 |
| **[프로그램 소스]**  #include <iostream>  #include <ctime>  using namespace std;  class Random {  public:  int select;  Random() { };  Random(int s) { select = s; }  int HollJack();  int HollJackRange(int a, int b);  };  int Random::HollJack()  {  int randNum;  if (select == 1) //홀수일때  {  do {  randNum = rand();  } while (randNum % 2 == 0);  return randNum;  }  else if (select == 2) //짝수일때  {  do {  randNum = rand();  } while (randNum % 2 == 1);  return randNum;  }  }  int Random::HollJackRange(int a, int b)  {  int randNum;  if (select == 1) //홀수일때  {  do {  randNum = rand() % (b - a + 1) + a;  } while (randNum % 2 == 0);  return randNum;  }  else if (select == 2) //짝수일때  {  do {  randNum = rand() % (b - a + 1) + a;  } while (randNum % 2 == 1);  return randNum;  }  }  int main()  {  srand((unsigned int)time(NULL));  Random r1(1); //홀수  Random r2(2); //짝수  cout << "--0에서 " << RAND\_MAX << "까지의 짝수 랜덤 정수 10 개 --" << endl;  for (int i = 0; i < 10; i++)  {  int n = r2.HollJack();  cout << n << ' ';  }    cout << endl << endl << "-- 2에서 " << "4 까지의 홀수 랜덤 정수 10 개 --" << endl;  for (int i = 0; i < 10; i++)  {  int n = r1.HollJackRange(2, 4);  cout << n << ' ';  }    return 0;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 153p 문제8번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Integer {  public:  int num;  string str;  Integer() { };  Integer(int i) { num = i; };  Integer(string s) { str = s; num = stoi(str); }  int get() { return num; };  void set(int n) { num = n; };  int isEven() {  if (num % 2 == 0)  return 1;  else  return 0; }  };  int main()  {  Integer n(30);  cout << n.get() << ' ';  n.set(50);  cout << n.get() << ' ';    Integer m("300");  cout << m.get() << ' ';  cout << m.isEven() << ' ';  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 154p 문제10번 |
| **[프로그램 소스]**  <Add.h>  #pragma once  class Add {  public:  int a;  int b;  Add() { a = 0; b = 0; }  void setValue(int x, int y);  int calculate();  };  <Sub.h>  #pragma once  class Sub {  public:  int a, b;    void setValue(int x, int y);  int calculate();  };  <Mul.h>  #pragma once  class Mul {  public:  int a, b;  Mul() { a = 0; b = 0; }  void setValue(int x, int y);  int calculate();  };  <Div.h>  #pragma once  class Div {  public:  int a, b;  Div() { a = 0; b = 0; }  void setValue(int x, int y);  int calculate();  };  <Add.cpp>  #include "Add.h"    void Add::setValue(int x, int y)  {  a = x;  b = y;  }  int Add::calculate()  {  return a + b;  }  <Sub.cpp>  #include "Sub.h"  void Sub::setValue(int x, int y)  {  a = x;  b = y;  }  int Sub::calculate()  {  return a - b;  }  <Mul.cpp>  #include "Mul.h"  void Mul::setValue(int x, int y)  {  a = x;  b = y;  }  int Mul::calculate() {  return a \* b;  }  <Div.cpp>  #include "Div.h"  void Div::setValue(int x, int y)  {  a = x;  b = y;  }  int Div::calculate()  {  return a / b;  }  <Calculator.cpp>  #include <iostream>  #include "Add.h"  #include "Sub.h"  #include "Mul.h"  #include "Div.h"  using namespace std;  int main()  {  int n1, n2;  char menu;  Add a; // 덧셈 객체 a 생성  Sub s; //뺄셈 객체 s 생성  Mul m; //곱셈 객체 m 생성  Div d; //나눗셈 객체 d 생성  while (1)  {  cout << "두 정수와 연산자를 입력하세요 >> ";  cin >> n1 >> n2 >> menu;  if (menu == '+')  {  a.setValue(n1, n2);  cout << a.calculate() << endl;  }  else if (menu == '-')  {  s.setValue(n1, n2);  cout << s.calculate() << endl;  }  else if (menu == '\*')  {  m.setValue(n1, n2);  cout << m.calculate() << endl;  }  else if (menu == '/')  {  d.setValue(n1, n2);  cout << d.calculate() << endl;  }  }  return 0;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 156p 문제12번 |
| **[프로그램 소스]**  <Main.cpp>  #include <iostream>  #include "Ram.h"  using namespace std;  int main()  {  Ram ram; //ram객체 생성  ram.write(100, 20); //100 번지에 20 저장  ram.write(101, 30); //101 번지에 30 저장  char res = ram.read(100) + ram.read(101);  ram.write(102, res);  cout << "102 번지의 값 = " << (int)ram.read(102) << endl;  return 0;  }  <Ram.h>  #pragma once  #include <iostream>  using namespace std;  class Ram {  char mem[100 \* 1024];  int size;  public:  Ram() {  mem[100 \* 1024] = { 0 }; //mem 배열을 0으로 초기화  size = 100 \* 1024; //sizefmf 100\*1024로 초기화  }  ~Ram() {  cout << "메모리가 제거됨" << endl;  }  char read(int address);  void write(int address, char value);  };  <Ram.cpp>  #include "Ram.h"  char Ram::read(int address)  {  return mem[address];  }  void Ram::write(int address, char value)  {  mem[address] = value;  } |
| **[실행결과]** |