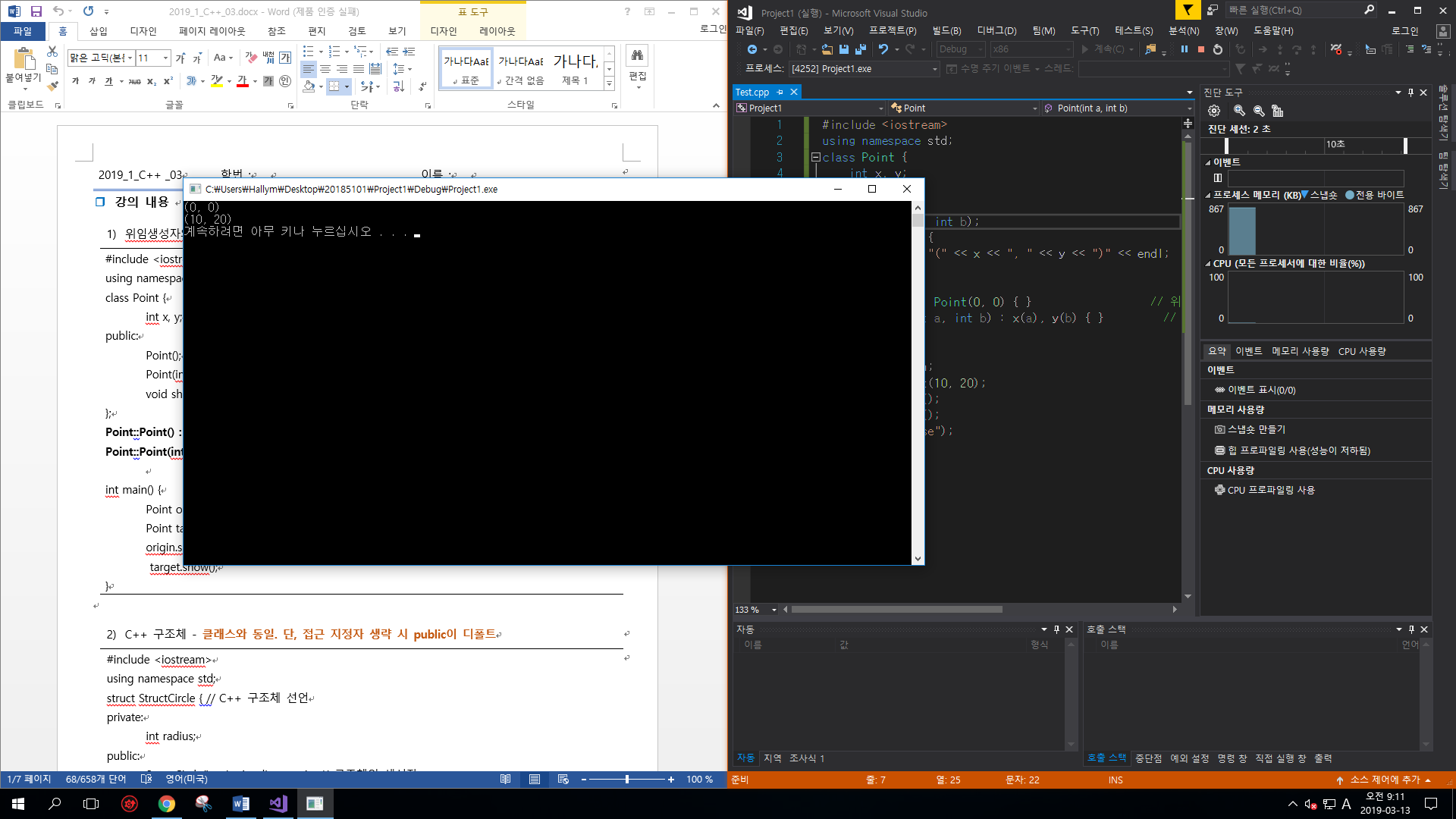
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_03 | 학번 : | 20145165 | 이름 : | 정균모 |

* **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** |
| #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 "Adder.h"  #include <iostream>  using namespace std;  int main(){  int op1, op2;  cout << "두 개의 수를 입력하세요>>";  cin >> op1 >> op2;  Adder adder(op1,op2);  cout << adder.process();  system("pause");  } |
| **[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 | 2 |
| 4 | Class Circle {  Int age;  Int radius;  Public:  Double getArea();  Void older(){  Age++;  }  } |
| 5 | Class Circle{  Int radius;  Double getArea();  }; |
| 6 | Class Tower{  Int height;  Public:  Tower() { height = 10; return ;}  }; |
| 7 | Class Building{  Private:  Int floor;  Public:  Building(int s) { floor = s; }  };  Int main(){  Building BlueHouse(5), JangMi(14);  } |
| 8 | Class Calender{  Private:  Int year;  Public:  Calender(year = 10;);  Int getYear() {return year;}  }; |
| 9 | 2 |
| 10 | 3: 소멸자는 반환값과 매개변수가 없다. |
| 11 | (1) public:  House(int n, int s) {  numOfRooms = n, size = s;  }  (2) ~House(){cout << numOfRooms << size};  (3) 생성 순서: bcad 소멸 순서: adcb |
| 12 | 생성 순서: c b a 소멸 순서: a b c |
| 13 | class TV {  public:  int channels;  TV() { channels = 256; } -> 생성자가 접근 지정자 안에 있어야 한다.  TV(int a) { channels = a; }  };  int main() {  TV LG;  LG.channels = 200;  } |
| 14 | class TV {  public:  int channels; -> channels 는 private라 접근이 불가하니 public 안으로 들어와야 한다.  int colors;  TV() { channels = 256; }  TV(int a) { channels = a; colors = b; }  };  int main() {  TV LG;  LG.channels = 200;  } |
| 15 | class TV {  int channels;  public:  TV() { channels = 256; } -> 선언부에 있는 함수는 자동으로 인라인  TV(int a) { channels = a; } -> 선언부에 있는 함수는 자동으로 인라인  }; |
| 16 | 2 |
| 17 | 1 |
| 18 | 1 |
| 19 | 4 |
| 20 | class Family {  char tel[11];  public:  int count;  char address[20];  Family();  }; |

* 프로그램 과제

|  |
| --- |
| 1. 교재 150p 문제2번   힌트) 문자열 길이를 반환하는 인스턴스 메소드 : lenth() |
| **[프로그램 소스]**  #include <string>  #include <iostream>  using namespace std;  class Date {  string independence;  int year, month, day;  public:  Date(int y, int m, int d) { year = y, month = m, day = d; }  Date(string str) { independence = str; }  void show();  int getYear();  int getMonth();  int getDay();  };  void Date::show() {  int year, month, day;  year = stoi(Date::independence.substr(0,4));  month = stoi(Date::independence.substr(5, 6));  day = stoi(Date::independence.substr(7, 8));  cout << year << "년" << month << "월" << day << "일" << endl;  }  int Date::getYear() {  return Date::year;  }  int Date::getMonth() {  return Date::month;  }  int Date::getDay() {  return Date::day;  }  int main() {  Date birth(2014, 3, 20);  Date independenceDay("1945/8/15");  independenceDay.show();  cout << birth.getYear() << ',' << birth.getMonth() << ',' << birth.getDay() << endl;  system("pause");  } |
| **[실행결과]** |

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

|  |
| --- |
| 1. 교재 152p 문제7번   힌트) 디폴트 생성자에서 time() 함수를 사용하여 시드값 설정 – 매번 다른 난수를 생성 |
| **[프로그램 소스]**  #include <iostream>  #include <cstdlib>  #include <ctime>  #include <string>  using namespace std;  class SelectableRandom {  string s;  public:  SelectableRandom(string s) { this->s = s; srand((unsigned)time(0)); }  int next();  int nextInRange(int a, int b);  string getstr() { return s; }  };  int SelectableRandom::next() {  int n = rand();  if (s == "홀수")  while (n % 2 == 0) n = rand();  else if (s == "짝수")  while (n % 2 == 1) n = rand();  return n;  }  int SelectableRandom::nextInRange(int a, int b) {  int n;  n = a + (rand() % (b - a + 1));  if (s == "홀수")  while (n % 2 == 0) n = a + (rand() % (b - a + 1));  else if (s == "짝수")  while (n % 2 == 1) n = a + (rand() % (b - a + 1));  return n;  }  int main() {  SelectableRandom r1("짝수");  SelectableRandom r2("홀수");  cout << "-- 0에서 " << RAND\_MAX << "까지의 " << r1.getstr() << " 랜덤 정수 10 개--" << endl;  for (int i = 0; i < 10; i++) {  int n = r1.next();  cout << n << ' ';  }  cout << endl << endl << "-- 2에서 " << "9 까지의 랜덤 " << r2.getstr() << " 정수 10 개 --" << endl;  for (int i = 0; i < 10; i++) {  int n = r2.nextInRange(2, 9);  cout << n << ' ';  }  cout << endl;  system("pause");  } |
| **[실행결과]** |

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

|  |
| --- |
| 1. 교재 154p 문제10번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Add {  int a;  int b;  public:  void setValue(int x, int y);  int calculate();  };  class Sub {  int a;  int b;  public:  void setValue(int x, int y);  int calculate();  };  class Mul {  int a;  int b;  public:  void setValue(int x, int y);  int calculate();  };  class Div {  int a;  int b;  public:  void setValue(int x, int y);  int calculate();  };  void Add::setValue(int x, int y) {  a = x; b = y;  }  int Add::calculate() {  return a + b;  }  void Sub::setValue(int x, int y) {  a = x; b = y;  }  int Sub::calculate() {  return a - b;  }void Mul::setValue(int x, int y) {  a = x; b = y;  }  int Mul::calculate() {  return a \* b;  }void Div::setValue(int x, int y) {  a = x; b = y;  }  int Div::calculate() {  return a / b;  }  int main() {  char cal[10]; char \*p;  int x, y;  while (1) {  cout << "두 정수와 연산자를 입력하세요>>";  cin.getline(cal, 20);  x = stoi(strtok(cal," "));  y = stoi(strtok(NULL," "));  p = strtok(NULL, "");  if (strcmp(p, "+") == 0) {  Add a;  a.setValue(x, y);  cout << a.calculate() << endl;  }  else if (strcmp(p, "-") == 0) {  Sub s;  s.setValue(x, y);  cout << s.calculate() << endl;  }  else if (strcmp(p, "\*") == 0) {  Mul m;  m.setValue(x, y);  cout << m.calculate() << endl;  }  else if (strcmp(p, "/") == 0) {  Div d;  d.setValue(x, y);  cout << d.calculate() << endl;  }  }  system("pause");  } |
| **[실행결과]** |

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