|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2018\_2\_C++ \_14 | 학번 : | 20147351 | 이름 : | 정지현 |

* setf(), unsetf()를 사용한 포맷 출력

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  cout << 30 << endl; // 10진수로 출력  cout.unsetf(ios::dec); // 10진수 해제  cout.setf(ios::hex); // 16진수로 설정  cout << 30 << endl;  cout.setf(ios::showbase); // 16진수로 설정  cout << 30 << endl;  cout.setf(ios::uppercase); // 16진수의 A~F는 대문자로 출력  cout << 30 << endl;  cout.setf(ios::dec | ios::showpoint); // 10진수 표현과 동시에 실수에 소숫점이하 나머지는 0으로 출력  cout << 23.5 << endl;  cout.setf(ios::scientific); // 실수를 과학산술용 표현으로 출력  cout << 23.5 << endl;  cout.setf(ios::showpos); // 양수인 경우 + 부호도 함께 출력  cout << 23.5;  } |
| **[실행 결과]** |

* width(), fill(), precision()을 사용한 포맷 출력

|  |
| --- |
| #include <iostream>  using namespace std;  void showWidth() {  cout.width(10); // 다음에 출력되는 "Hello"를 10 칸으로 지정  cout << "Hello" << endl;  cout.width(5); // 다음에 출력되는 정수 12를 5 칸으로 지정  cout << 12 << endl;  cout << '%';  cout.width(10); // 다음에 출력되는 "Korea/"만 10 칸으로 지정  cout << "Korea/" << "Seoul/" << "City" <<endl;  }  int main() {  showWidth(); // width() 사용 사례  cout << endl;  cout.fill('^'); // fill()을 적용한 후 width()의 사례를 보여준다.  showWidth();  cout << endl;  cout.precision(5); // precision() 사용 예  cout << 11./3. << endl;  } |
| **[실행 결과]** |

* 매개 변수를 가진 조작자

|  |
| --- |
| #include <iostream>  #include <iomanip>  using namespace std;  int main() {  cout << showbase;  // 타이틀을 출력한다.  cout << setw(8) << "Number";  cout << setw(10) << "Octal";  cout << setw(10) << "Hexa" << endl;  // 하나의 수를 십진수, 8진수, 16진수 형태로 한 줄에 출력한다.  for(int i=0; i<50; i+=5) {  cout << setw(8) << setfill('.') << dec << i; // 10진수  cout << setw(10) << setfill(' ') << oct << i; // 8진수  cout << setw(10) << setfill(' ') << hex << i << endl; // 16진수  }  } |
| **[실행 결과]** |

* Point 객체를 입출력 하는 >>, << 연산자 작성

|  |
| --- |
| #include <iostream>  using namespace std;  class Point { // 한 점을 표현하는 클래스  int x, y; // private 멤버  public:  Point(int x=0, int y=0) {  this->x = x;  this->y = y;  }  friend istream& operator >> (istream& ins, Point &a); // friend 선언  friend ostream& operator << (ostream& stream, Point a); // friend 선언  };  istream& operator >> (istream& ins, Point &a) { // >> 연산자 함수  cout << "x 좌표>>";  ins >> a.x;  cout << "y 좌표>>";  ins >> a.y;  return ins;  }  ostream& operator << (ostream& stream, Point a) { // << 연산자 함수  stream << "(" << a.x << "," << a.y << ")";  return stream;  }  int main() {  Point p; // Point 객체 생성  cin >> p; // >> 연산자를 호출하여 x 좌표와 y 좌표를 키보드로 읽어 객체 p 완성  cout << p; // << 연산자를 호출하여 객체 p 출력  } |
| **[실행 결과]** |

* 내용 점검

1. 교재 11장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | 3. |
| 2 | 4. |
| 3 | 키보드 |
| 4 | 스크린 |
| 5 | cin, cout, clog, cerr |
| 6 | 2. |
| 7 | 1. |
| 8 | char |
| 9 | abc => put() 함수는 연결하여 사용가능 |
| 10 | get() 함수는 2가지 종류가 있는데 2바이트로 구성되는 한글 문자를 읽을 수 없으며 입력 스트림에서 한 개의 문자를 읽어 리턴해주고 <enter>키 등의 공백 문자도 읽어 리턴한다. 또한 get()함수의 리턴 타입은 int이다. 또한 버퍼에 그대로 둔다. getline() 함수는 지정된 구분 문자를 만날 때까지, 혹은 n-1개의 문자를 모두 읽을 때까지 배열의 문자를 읽어 들이는 함수이다. 또한 스트림 버퍼에서 제거가 가능하다. 스트림 버퍼에서 제거하지 않으면 무한루프에 빠질 수 있다. |
| 11 | 1. |
| 12 | 15 |
| 13 | 2,3,4. |
| 14 | 4. |
| 15 | 4. |
| 16 | %%%%%%%%%%%%C++ |
| 17 | 0.6667~~~~ |
| 18 | <iostream>, <iomanip> |
| 19 | <ostream> 헤더파일, 원형은 ostream& operator << (char); |
| 20 | 4. |

* 프로그램 과제

1. 교재 581p : 문제 4번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  #include<string>  using namespace std;  int main() {  char ch;  while ((ch = cin.get()) != EOF) {  if (ch != ';')  cout << (char)ch;    if (ch == ';') {  cout << endl;  cin.ignore(100, '\n');    }  }  char ch1;  ch1 = getchar();  } |
| **[실행 결과]** |

1. 교재 583p : 문제 7번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <iomanip>  using namespace std;  int main(){  cout << showbase;  for (int i = 0; i < 2; i++) {  cout << setw(10) << "dec";  cout << setw(10) << "hexa";  cout << setw(10) << "char";  }  // 하나의 수를 십진수, 8진수, 16진수 형태로 한 줄에 출력한다.  for (int i = 0; i < 128; i++) {  if (i % 2 == 0) {  cout << endl;  }  cout << setw(10) << dec << i; // 10진수  cout << setw(10) << hex << i; // 8진수  if (isprint(char(i))) {  cout << setw(10) << (char)i; // 16진수  }  else {  cout << setw(10) << '.';  }    }  char ch;  ch = getchar();  } |
| **[실행 결과]** |

1. 교재 584p : 문제 9번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  #include<string>  using namespace std;  class Phone {  string name;  string telnum;  string address;  public:  Phone(string name = "", string telnum = "", string address = "") {  this->name = name;  this->telnum = telnum;  this->address = address;  }  friend istream& operator >> (istream& ins, Phone &a);  friend ostream& operator << (ostream& stream, Phone a);  };  istream& operator >> (istream& ins, Phone &a) {  cout << "이름 >>";  ins >> a.name;  cout << "전화번호 >>";  ins >> a.telnum;  cout << "주소 >> ";  ins >> a.address;  return ins;  }  ostream& operator << (ostream& stream, Phone a) {  stream << "(" << a.name << "," << a.telnum << "," << a.address << ")";  return stream;  }  int main() {  Phone girl, boy;  cin >> girl >> boy;  cout << girl << endl << boy << endl;  char ch;  ch = getchar();  ch = getchar();  } |
| **[실행 결과]** |

1. 교재 586p : 문제 12번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Ingredient {  protected:  string name;  int amount;  bool isEmpty;  public:  Ingredient() { isEmpty = false; }  void decreaseAmount() {  amount = amount - 1; // 잔여량을 1개 감소시킨다.  if (amount == 0) isEmpty = true; // 잔여량이 0개이면 빈 것  }  int getAmount() { return amount; }  void setAmount(int amount) { this->amount = amount; }  string getName() { return name; }  bool getIsEmpty() { return isEmpty; }  void setIsEmpty(bool isEmpty) { this->isEmpty = isEmpty; }  };  class Coffee : public Ingredient {  public:  Coffee() { this->name = "Coffee"; this->amount = 3; } // 이름과 잔여량 초기화  };  class Sugar : public Ingredient {  public:  Sugar() { this->name = "Sugar"; this->amount = 3; } // 이름과 잔여량 초기화  };  class Cream : public Ingredient {  public:  Cream() { this->name = "Cream"; this->amount = 3; } // 이름과 잔여량 초기화  };  class Water : public Ingredient {  public:  Water() { this->name = "Water"; this->amount = 3; } // 이름과 잔여량 초기화  };  class Cup : public Ingredient {  public:  Cup() { this->name = "Cup"; this->amount = 3; } // 이름과 잔여량 초기화  };  class VendingMachine {  Ingredient \*ingredients[5]; // Ingredient형 포인터 객체 배열 선언  public:  VendingMachine() {  cout << "------명품 커피 자판기켭니다.------" << endl;  // 0: Coffee, 1: Sugar, 2: Cream, 3: Water, 4:Cup 으로 초기화  ingredients[0] = new Coffee();  ingredients[1] = new Sugar();  ingredients[2] = new Cream();  ingredients[3] = new Water();  ingredients[4] = new Cup();  }    void makeCoffee(int type);  bool checkEmpty(int type);    void run();  int selectMenu();    void refillIngredients();  void showIngredients();  };  bool VendingMachine::checkEmpty(int type) {  // 공통 재료 0(커피), 3(물), 4(컵) 이 없는 경우 체크  if (ingredients[0]->getIsEmpty() || ingredients[3]->getIsEmpty() ||  ingredients[4]->getIsEmpty())  return true;  switch (type) {  case 0:  break;    case 1:  return ingredients[1]->getIsEmpty(); // 설탕 없는 경우 체크, 리턴    case 2:  return ingredients[2]->getIsEmpty(); // 크림 없는 경우 체크, 리턴  }    return false; // 부족한 재료가 없는 상태임을 리턴  }  void VendingMachine::makeCoffee(int type) {  if (checkEmpty(type))  cout << "재료가 부족합니다." << endl;  else {  ingredients[0]->decreaseAmount(); // 커피 잔여량 감소  ingredients[3]->decreaseAmount(); // 물 잔여량 감소  ingredients[4]->decreaseAmount(); // 컵 잔여량 감소  switch (type) {  case 0:  cout << "맛있는 보통 커피 나왔습니다." << endl; break;    case 1:  cout << "맛있는 설탕 커피 나왔습니다." << endl;  ingredients[1]->decreaseAmount(); // 설탕 잔여량 감소  break;  case 2:  cout << "맛있는 블랙 커피 나왔습니다." << endl;  ingredients[2]->decreaseAmount(); // 크림 잔여량 감소  break;  default: cout << "잘못 선택하셨습니다." << endl;  }  }  }  void VendingMachine::run()  {  while (true) {  showIngredients(); // 메뉴 잔여량 출력    int menu = selectMenu();  switch (menu) {  case 0:  case 1:  case 2:  makeCoffee(menu); break;  case 3:  refillIngredients(); break;  case 4:  return;  }  }  }  int VendingMachine::selectMenu()  {  cout << endl;  cout << "보통 커피:0, 설탕 커피:1, 블랙 커피:2, 채우기:3, 종료:4 >>";  int choiceMenu = 0;  cin >> choiceMenu;  return choiceMenu;  }  void VendingMachine::refillIngredients()  {  cout << "모든 통을 채웁니다.~~" << endl;  for (int idx = 0; idx < 5; idx++) {  ingredients[idx]->setAmount(3); // setter를 이용한 잔여량 조정  ingredients[idx]->setIsEmpty(false); // setter를 이용한 재료 잔여 여부 변경  }  }  void VendingMachine::showIngredients()  { // for문을 이용한 모든 재료의 이름 및 잔여량 출력  for (int idx = 0; idx < 5; idx++) {  cout << ingredients[idx]->getName() << " : ";  for (int amount = 0; amount < ingredients[idx]->getAmount(); amount++) {  cout << "\*";  }  cout << endl;  }  }  int main(void)  {  VendingMachine vm;  vm.run();  return 0;  char ch;  ch = getchar();  } |
| **[실행 결과]** |