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

* 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 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |
| 18 |  |
| 19 |  |
| 20 |  |

* 프로그램 과제

1. 교재 581p : 문제 4번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  int main() {  int ch;  while((ch=cin.get()) != EOF) {  if(ch == ';') {  cin.ignore(100, '\n'); // 한글 문장이 최대 99개의 문자로 입력된다고 가정한다.  cout << '\n';  }  else  cout << (char)ch;  }  } |
| **[실행 결과]** |

1. 교재 583p : 문제 7번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <cctype>  using namespace std;  int main() {  cout << "dec" << '\t' << "hexa" << '\t' << "char" << '\t';  cout << "dec" << '\t' << "hexa" << '\t' << "char" << '\t';  cout << "dec" << '\t' << "hexa" << '\t' << "char" << '\t';  cout << "dec" << '\t' << "hexa" << '\t' << "char" << endl;  cout << "\_\_\_" << '\t' << "\_\_\_\_" << '\t' << "\_\_\_\_" << '\t';  cout << "\_\_\_" << '\t' << "\_\_\_\_" << '\t' << "\_\_\_\_" << '\t';  cout << "\_\_\_" << '\t' << "\_\_\_\_" << '\t' << "\_\_\_\_" << '\t';  cout << "\_\_\_" << '\t' << "\_\_\_\_" << '\t' << "\_\_\_\_" << endl;  cout.width(3);  cout << left;  for(int i=0; i<128; i++) {  cout << dec << i << '\t' << hex << i << '\t' << (isprint(i)?(char)i:'.') << '\t';  if(i%4 == 3)  cout << endl;  }  } |
| **[실행 결과]** |

1. 교재 584p : 문제 9번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Phone { // 전화 번호를 표현하는 클래스  string name;  string telnum;  string address;  public:  Phone(string name="", string telnum="", string address="") {  this->name = name;  this->telnum = telnum;  this->address = address;  }  friend ostream& operator << (ostream& outs, Phone phone); // << 연산자를 프렌드로 선언  friend istream& operator >> (istream& ins, Phone& phone); // >> 연산자를 프렌드로 선언  };  ostream& operator << (ostream& outs, Phone phone) { // 스트림에 phone 객체 출력  outs << "(" << phone.name << ",";  outs << phone.telnum << ",";  outs << phone.address << ")";  return outs;  }  istream& operator >> (istream& ins, Phone& phone) { // 스트림으로부터 읽어 phone 객체 완성  cout << "이름:";  getline(ins, phone.name);  cout << "전화번호:";  getline(ins, phone.telnum);  cout << "주소:";  getline(ins, phone.address);  return ins;  }  int main() {  Phone girl, boy;  cin >> girl >> boy; //전화 번호를 키보드로부터 읽는다.  cout << girl << endl << boy << endl; // 전화 번호를 출력한다.  } |
| **[실행 결과]** |

1. 교재 586p : 문제 12번

|  |
| --- |
| **[프로그램 소스]**  #ifndef CONTAINER\_H  #define CONTAINER\_H  #include <string>  using namespace std;  #include "define.h"  class Container { // 통을 표현하는 클래스  string name;  int size; // 통의 크기  public:  Container(const char\* name="", int initialSize=MAXSIZE);  bool consume(int n=1); // 재료 소비  void fill() { size = MAXSIZE; } // 재료 채우기  int getSize() { return size; } // 현재 남은 재료의 량 리턴  bool isEmpty() { return size==0?true:false; }  bool isFull() { return size==MAXSIZE?true:false; }  void put() { size++; } // 재료 량 1 증가  void setName(string name) { this->name = name; }  void show(); // 통의 상황 출력  };  #endif  ===========================================  // 프로그램 전체에 필요한 상수를 선언하는 파일  #ifndef DEFINE\_H  #define DEFINE\_H  #define MAXSIZE 3 // 통의 최대 크기  enum { COFFEE, SUGAR, CREAM, WATER, CUP }; // 통의 종류  enum { NORMAL\_COFFEE, SUGAR\_COFFEE, BLACK\_COFFEE, FILL, EXIT}; //버튼 종류  #endif  =======================================================================  #ifndef ENGINE\_H  #define ENGINE\_H  #include <string>  using namespace std;  #include "Container.h"  // 커피 자판기의 핵심 클래스로서, 사용자의 커피 선택으로부터  // 각 통에 들어 있는 재료를 줄이거나 늘리는 등,  // 커피 자판기를 시뮬레이션 하는 클래스  class Engine {  Container \*con; // 전체 통들에 대한 포인터  public:  Engine(Container []);  void run(); // 엔진 작동  void show();  void printWarning(int id);  };  #endif  =====================================================================  #ifndef PANEL\_H  #define PANEL\_H  // 커피 자판기의 패널(커피 선택 버튼 등)을 시뮬레이션하는 클래스  class Panel {  public:  static int getButton(); // 커피 메뉴를 출력하고 사용자로부터 커피 메뉴 입력받는 함수  };  #endif  ======================================================================  #ifndef VENDINGMACHINE\_H  #define VENDINGMACHINE\_H  #include <string>  using namespace std;  class Engine;  class Container;  class VendingMachine { // 커피 자판기 전체를 관리하는 클래스  Container \*con; // 커피, 물 등의 통들을 배열로 관리하기 위한 포인터  string name; // 자판기 이름  Engine \*engine; // 사용자의 커피 종류 선택에서 부터 자판기를 시뮬레이션하는 엔진에 대한 포인터  public:  VendingMachine(string name);  ~VendingMachine();  void powerOn(); // 커피 자판기 시작. 이 함수가 끝나면 커피 자판기 종료  };  #endif  ===========================================================================  #include <iostream>  using namespace std;  #include "Container.h"  Container::Container(const char\* name, int initialSize) {  this->name = name;  size = initialSize;  }  bool Container::consume(int n) { // 통에서 n 분만큼 재료 소모  if(size < n) return false;  size -= n;  return true;  }  void Container::show() { // 통에 현재 남은 재료 량을 '\*'로 출력  cout << name + "\t";  for(int i=0; i<size; i++)  cout << "\*";  cout << endl;  }  ===========================================================================  #include <iostream>  using namespace std;  #include "define.h"  #include "Container.h"  #include "Engine.h"  #include "Panel.h"  Engine::Engine(Container con[]) { // 커피 자판기에 존재하는 통들에 대한 포인터 전달 받음  this->con = con;  }  void Engine::run() { // 엔진 작동  show(); // 현재 커피 자판기 상태 출력. 현재 각 통의 용량 출력  while(true) {  int menu = Panel::getButton(); // 사용자로부터 커피 종류 선택하여 리턴  switch(menu) {  case BLACK\_COFFEE :  if(!con[COFFEE].consume()) { // 커피 줄임  printWarning(COFFEE);  break;  }  if(!con[WATER].consume()) { // 물 줄임  printWarning(WATER);  break;  }  if(!con[CUP].consume()) { // 컵 줄임  printWarning(CUP);  break;  }  cout << "맛있는 블랙 커피 나왔습니다~~" << endl;  break;  case SUGAR\_COFFEE :  if(!con[COFFEE].consume()) { // 커피 줄임  printWarning(COFFEE); // 커피가 없을 때 경고 메시지 출력  break;  }  if(!con[WATER].consume()) { // 물 줄임  printWarning(WATER);  break;  }  if(!con[CUP].consume()) { // 컵 줄임  printWarning(CUP);  break;  }  if(!con[SUGAR].consume()) { // 설탕 줄임  printWarning(SUGAR);  break;  }  cout << "맛있는 설탕 커피 나왔습니다~~" << endl;  break;  case NORMAL\_COFFEE:  if(!con[COFFEE].consume()) { // 커피 줄임  printWarning(COFFEE);  break;  }  if(!con[WATER].consume()) { // 물 줄임  printWarning(WATER);  break;  }  if(!con[CUP].consume()) { // 컵 줄임  printWarning(CUP);  break;  }  if(!con[SUGAR].consume()) { // 설탕 줄임  printWarning(SUGAR);  break;  }  if(!con[CREAM].consume()) { // 크림 줄임  printWarning(CREAM);  break;  }  cout << "맛있는 다방 커피 나왔습니다~~" << endl;  break;  case FILL: // 모든 재료를 최대로 채운다.  cout << "모든 통을 채웁니다.~~" << endl;  for(int i=0; i<5; i++) con[i].fill(); // 모든 통에 재료 최대로 채움  break;  case EXIT: // 종료 메뉴  return; // 엔진 종료  }  show();  }  }  void Engine::show() {  for(int i=0; i<5; i++)  con[i].show();  cout << endl;  }  void Engine::printWarning(int id) {  switch(id) {  case COFFEE : cout << "커피가 없습니다." <<endl; break;  case SUGAR : cout << "설탕이 없습니다." <<endl; break;  case CREAM : cout << "크림이 없습니다." <<endl; break;  case WATER : cout << "물이 없습니다." <<endl; break;  case CUP : cout << "컵이 없습니다." <<endl; break;  }  }  =======================================================================================  #include "VendingMachine.h"  int main() {  VendingMachine machine("명품 커피 자판기");  machine.powerOn(); // 커피 자판기 작동 시작  }  ==========================================================================================  #include <iostream>  using namespace std;  #include "define.h"  #include "Panel.h"  int Panel::getButton() {  int menu;  cout <<"보통 커피:"<< NORMAL\_COFFEE << ", ";  cout <<"설탕 커피:"<< SUGAR\_COFFEE << ", ";  cout <<"블랙 커피:"<< BLACK\_COFFEE << ", ";  cout <<"채우기:"<< FILL << ", ";  cout << "종료:" << EXIT << ">> ";  cin >> menu;  return menu;  }  ======================================================================================  #include <iostream>  #include <string>  using namespace std;  #include "define.h"  #include "Engine.h"  #include "VendingMachine.h"  VendingMachine::VendingMachine(string name) {  this->name = name;  con = new Container[5]; // 통 생성  con[COFFEE].setName("Coffee");  con[SUGAR].setName("Sugar");  con[CREAM].setName("CREAM");  con[WATER].setName("Water");  con[CUP].setName("Cup");  engine = new Engine(con); // 엔진 생성  }  VendingMachine::~VendingMachine() {  delete engine; // 엔진 소멸  delete [] con; // 생성한 모든 통 객체 소멸  }  void VendingMachine::powerOn() { // 자판기가 켜질 때를 시뮬레이션하는 함수  cout << "------" << name << "켭니다.------" << endl;  engine->run(); // 엔진 시작 시킴  } |
| **[실행 결과]** |