|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_14 | 학번 : | 20145128 | 이름 : | 박인근 |

* 내용 점검 – 교재11장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | 3 |
| 2 | 4 |
| 3 | 키보드 |
| 4 | 모니터(스크린) |
| 5 | cin, cout, clog, cerr |
| 6 | 2 |
| 7 | 1 |
| 8 | char |
| 9 | abc 출력  이유는 put() 함수는 ostream& 타입이여서 cout의 참조를 반환하기 때문에 a 가 먼저 cout 스트림에 삽입되고 다음에 b는 다시 cout 버퍼에 삽입됩니다. 따라서 순서대로 출력됩니다. |
| 10 | get() 은 ‘\n’ 을 입력 스트림 버퍼에 남겨 두지만, getline() 은 ‘\n’ 을 남겨 두지 않고 제거합니다. |
| 11 | 1  띄어쓰기 이후에는 저장되지 않습니다. |
| 12 | 15  ‘\n’ 을 남겨 두지 않고 제거하기 때문에 <Enter> 키도 읽은 문자 개수에 포함됩니다. |
| 13 | 2,3 |
| 14 | 4 |
| 15 | 4 |
| 16 | %%%%%%%%%C++ |
| 17 | 0.6667~~~~ |
| 18 | <iostream> , <iomanip> |
| 19 | <ostream> 헤더 파일에 포함되어 있습니다.  함수의 원형은 ostream& operator << (char); 입니다. |
| 20 | 4 |

* 프로그램 과제

|  |
| --- |
| 1. 교재 583p : 문제7번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  #include <cctype>  #include <iomanip>  using namespace std;  int main()  {  cout << noshowbase;  for (int i = 0; i < 4; i++) {  cout << setw(10) << left << "dec";  cout << setw(10) << left << "hexa";  cout << setw(10) << left << "char";  }  cout << endl;  for (int i = 0; i < 4; i++) {  cout << setw(10) << left << "---";  cout << setw(10) << left << "----";  cout << setw(10) << left << "----";  }  for (int i = 0; i < 128; i++) {  if (i % 4 == 0) cout << endl;  if (isprint(i)) {  cout << setw(10) << left << dec << i;  cout << setw(10) << left << hex << i;  cout << setw(10) << left << char(i);  }  else {  cout << setw(10) << left << dec << i;  cout << setw(10) << left << hex << i;  cout << setw(10) << left << '.';  }  }    } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 584p : 문제9번 |
| **[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:";  getline(ins, a.name);  cout << "전화번호:";  getline(ins, a.telnum);  cout << "주소:";  getline(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;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 585p : 문제10번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  istream& prompt(istream& ins) {  cout << "암호?";  return ins;  }  int main()  {  string password;  while (true) {  cin >> prompt >> password;  if (password == "C++") {  cout << "login success!!" << endl;  break;  }  else  cout << "login fail. try again!!" << endl;  }  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 586p : 문제12번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class MachineBase {  protected:  string name;  int count;  bool empty;  public:  MachineBase() {  empty = false;  }  void useCount() {  count -= 1;  if (count == 0) {  empty = true;  }  }  int getCount() { return count; }  void setCount(int count) { this->count = count; }  string getName() { return name; }  bool getEmpty() { return empty; }  void setEmpty(bool empty) { this->empty = empty; }  };  class Coffee : public MachineBase {  public:  Coffee() {  this->name = "Coffee";  this->count = 3;  }  };  class Sugar : public MachineBase {  public:  Sugar() {  this->name = "Sugar";  this->count = 3;  }  };  class Cream : public MachineBase {  public:  Cream() {  this->name = "Cream";  this->count = 3;  }  };  class Water : public MachineBase {  public:  Water() {  this->name = "Water";  this->count = 3;  }  };  class Cup : public MachineBase {  public:  Cup() {  this->name = "Cup";  this->count = 3;  }  };  class coffeeMachine {  int select;  MachineBase \*MachineBase[5];  public:  coffeeMachine(){  cout << "------ 명품 커피 자판기 켭니다.------" << endl;  MachineBase[0] = new Coffee();  MachineBase[1] = new Sugar();  MachineBase[2] = new Cream();  MachineBase[3] = new Water();  MachineBase[4] = new Cup();  show();  }    void show() {  for (int i = 0; i < 5; i++) {  cout << MachineBase[i]->getName() << "\t";  for (int j = 0; j < MachineBase[i]->getCount(); j++) {  cout << "\*";  }  cout << endl;  }  }  void MachineRun() {  while (1) {  cout << "보통 커피:0, 설탕 커피:1, 블랙 커피:2, 채우기:3, 종료:4>> ";  cin >> select;  if (select == 3) {  cout << "모든 통을 채웁니다.~~" << endl;  for (int i = 0; i < 5; i++) {  MachineBase[i]->setCount(3);  MachineBase[i]->setEmpty(false);  }  show();  continue;  }  else if (select == 4) {  cout << "커피 자판기를 종료합니다!" << endl;  break;  }  if (MachineBase[0]->getEmpty() || MachineBase[1]->getEmpty() || MachineBase[2]->getEmpty() || MachineBase[3]->getEmpty() || MachineBase[4]->getEmpty()) {  cout << "재료가 부족합니다." << endl;  cout << "채우기 후 다시 시도해 주세요." << endl;  continue;  }  switch (select)  {  case 0:  MachineBase[0]->useCount();  MachineBase[3]->useCount();  MachineBase[4]->useCount();    cout << "맛있는 보통 커피 나왔습니다~~" << endl;  show();  break;    case 1:  MachineBase[0]->useCount();  MachineBase[1]->useCount();  MachineBase[3]->useCount();  MachineBase[4]->useCount();    cout << "맛있는 설탕 커피 나왔습니다~~" << endl;  show();  break;    case 2:  MachineBase[0]->useCount();  MachineBase[2]->useCount();  MachineBase[4]->useCount();    cout << "맛있는 블랙 커피 나왔습니다~~" << endl;  show();  break;  }  }  }    };  int main()  {    coffeeMachine machine;  machine.MachineRun();  system("pause");  } |
| **[실행결과]** |