|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_02 | 학번 : |  | 이름 : |  |

* **강의 내용**

|  |
| --- |
| 1. namesapce와 문자열 |
| #include <iostream>  #include <string>  #include <cstring>  using namespace std;  namespace A {  void input() {  char name1[20]; //c-스트링 사용  cout << "namespace A ====" << endl;  cout << "비교하고 싶은 문자열을 입력하세요 : ";  cin.getline(name1, 20);  if(strcmp(name1, "namespace")== 0)  cout << "동일한 문자열 "<<endl;  else  cout << "다른 문자열 " << endl;  }  }  namespace B {  void input() {  string name1, name2; //string 객체 사용  cout << "namespace B ====" << endl;  cout << "비교하고 싶은 문자열을 입력하세요 : ";  getline(cin, name1);  if (name1 == "name space")  cout << "동일한 문자열" << endl;  else  cout << "다른 문자열 " << endl;  }  }  int main() {  A::input();  B::input();  return 0;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 공백이 포함된 문자열 입력  * C스트링 : cin.getline(char buf[], int size, char delimitChar) |
| #include <iostream>  using namespace std;  int main() {  cout << "주소를 입력하세요>>";  char address[100];  // 키보드로부터 공백이 포함된 문자열 입력  cin.getline(address, 100, '\n');  cout << "주소는 " << address << "입니다\n";  } |
| **[실행결과]** |

|  |
| --- |
| 1. 공백이 포함된 문자열 입력  * string 클래스 : getline() |
| #include <iostream>  #include <string> //string 클래스를 사용하기 위한 헤더 파일  using namespace std; //std안에 string 정의, 이름영역지정 반드시 필요  int main() {  string song("Falling in love with you"); // 문자열 song  string elvis("Elvis Presley"); // 문자열 elvis  string singer; // 문자열 singer  cout << song + "를 부른 가수는"; // + 로 문자열 연결  cout << "(힌트 : 첫 글자는 " << elvis[0] << ")?"; // [] 연산자 사용  // getline()은 string 타입의 문자열을 입력 받기 위해 제공되는 전역 함수  // 공백이 포함된 문자열 입력 가능  // string을 사용하는 문자열은 c에서 사용하는 문자열 처리 함수 사용불가  getline(cin, singer); // 문자열 입력,  if(singer == elvis) // 문자열 비교  cout << "맞았습니다.";  else  cout << "틀렸습니다. "+ elvis + "입니다." << endl; // +로 문자열 연결  } |
| **[실행결과]** |

|  |
| --- |
| 1. Circle 클래스의 객체 생성 및 활용 |
| #include <iostream>  using namespace std;  class Circle { //Circle 선언부  public:  int radius;  double getArea();  };  double Circle::getArea() { //Circlee 구현부  return 3.14\*radius\*radius;  }  int main() {  Circle donut; //객체 생성  donut.radius = 1; // donut 객체의 반지름을 1로 설정  double area = donut.getArea(); // donut 객체의 면적 알아내기  cout << "donut 면적은 " << area << endl;  Circle pizza;  pizza.radius = 30; // pizza 객체의 반지름을 30으로 설정  area = pizza.getArea(); // pizza 객체의 면적 알아내기  cout << "pizza 면적은 " << area << endl;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 프로그램 분할 |
| **//Circle.h**  class Circle { //Circle 선언부  public:  int radius;  double getArea();  };  **//Circle.cpp**  #include "Circle.h"  double Circle::getArea() { //Circlee 구현부  return 3.14\*radius\*radius;  }  **//Main.cpp**  #include <iostream>  #include "Circle.h"  using namespace std;  int main() {  Circle donut; //객체 생성  donut.radius = 1; // donut 객체의 반지름을 1로 설정  double area = donut.getArea(); // donut 객체의 면적 알아내기  cout << "donut 면적은 " << area << endl;  Circle pizza;  pizza.radius = 30; // pizza 객체의 반지름을 30으로 설정  area = pizza.getArea(); // pizza 객체의 면적 알아내기  cout << "pizza 면적은 " << area << endl;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 소멸자  * 객체가 소멸될 때 실행 * 반환값과 매개변수 없음 * 객체 생성 순서와 반대로 소멸 |
| #include <iostream>  using namespace std;  class Circle {  public:  int radius;  Circle();  Circle(int r);  ~Circle(); //소멸자 선언  double getArea();  };  Circle::Circle() {  radius = 1;  cout << "반지름 " << radius << " 원 생성" << endl;  }  Circle::Circle(int r) {  radius = r;  cout << "반지름 " << radius << " 원 생성" << endl;  }  Circle::~Circle() { //소멸자 구현  cout << "반지름 " << radius << " 원 소멸" << endl;  }  double Circle::getArea() {  return 3.14\*radius\*radius;  }  Circle globalDonut(1000); //전역객체  Circle globalPizza(2000);  void f() {  Circle fDonut(100); //지역객체  Circle fPizza(200);  }  int main() {  Circle mainDonut; //지역객체  Circle mainPizza(30);  f();  } |
| **[실행결과]** |

* **내용 점검**

1. 교재 2장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | Int main() |
| 2 | 3번 -> 생략가능 |
| 3 | Return 0; |
| 4 | 1. C는 선언부가 아닌 곳에서 선언을 하면 안 된다. 2. 어디에서나 변수를 지정할 수 있는 특성 3. 장점: 변수를 상용하는 코드 바로 위에 변수를 선언할 수 있어 코드를 읽기 쉽다.   단점: 선언된 변수들이 코드 사이에 흩어져 있으므로, 선언된 모든 변수를 한 눈에 보기 힘들고, 코드 사이에서 선언된 변수를 찾기가 용이하지 않다. |
| 5 | I love C++  I love programming |
| 6 | (1)using std::cout;  (2)using namespace std; |
| 7 | (1)#include <iostream> (2)using namespace std; (3)std::cin >> name; (4) std::cout << 1 << 2 |
| 8 | (1)없다 (2)없다. (3)int n=1; cout << n+200 (4)int year=2014; cout<< 2014<<”년”; |
| 9 | #include “myheader.h” |
| 10 | (1)o (2)o (3)o (4)x (5)x |
| 11 | #include <cstring> |
| 12 | (1)Kitae님 환영합니다.  (2) Kitae님 환영합니다. |
| 13 | 4번 |
| 14 | 1번 |
| 15 | namespace |
| 16 | Std |
| 17 | Std |
| 18 | Iostream |
| 19 | cout << age << ' ' << pDept; |
| 20 | for (int j = 0; j <= n; j++) { cout << "\*"; }  cout << endl; |

* 프로그램 과제

1. 교재 91p 문제 9번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  int main() {  string name;  string addr;  int age;  cout << "이름을 입력하세요: ";  getline(cin, name);  cout << "주소를 입력하세요: ";  getline(cin, addr);  cout << "나이를 입력하세요: ";  cin >> age;  cout << name << ", " << addr << ", " << age << "세" << endl;  system("pause");  return 0;  } |
| **[실행 결과]** |

1. 교재 91p 문제 10번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  int main() {  int i,j;  char hello[6];  cout << "문자열 입력>>";  cin.getline(hello, 6, '\n');  for (i = 0; i < 5; i++) {  for (j = 0; j <= i; j++) {  cout << hello[j];  }  cout << endl;  }  system("pause");  } |
| **[실행 결과]** |

1. 교재 92p 문제 13번 – 출력 시 각 메뉴의 가격도 출력하며 주문 받은 메뉴에 대한 총 가격도 계산하여 출력할 수 있도록 하시오

메뉴만이 아니라 가격도 같이 해야 된다. 총 주문가격도 출력

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  int main() {  string menu = "짬뽕:1(500원), 짜장:2(400원), 군만두:3(300원), 종료:4>> ";  int choose,size;  bool open = true;  cout << "\*\*\*\*\* 승리장에 오신 것을 환영합니다. \*\*\*\*\*" << endl;  while (open) {  cout << menu;  cin >> choose;  switch (choose){  case 1:  cout << "몇인분?";  cin >> size;  cout << "짬뽕 " << size << "인분 나왔습니다." << endl << "가격은 :" << 500 \* size << "입니다." << endl;  break;  case 2:  cout << "몇인분?";  cin >> size;  cout << "짜장 " << size << "인분 나왔습니다." << endl << "가격은 :" << 400 \* size << "입니다." << endl;  break;  case 3:  cout << "몇인분?";  cin >> size;  cout << "군만두 " << size << "인분 나왔습니다." << endl << "가격은 :" << 300 \* size << "입니다." << endl;  break;  case 4:  cout << "오늘 영업은 끝났습니다." << endl;  open = false;  break;  default:  cout << "다시 입력하세요!!" << endl;  }  }  system("pause");  } |
| **[실행 결과]** |

1. 교재 92p 문제 14번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  int main() {  string menu = "에스프레소 2000원, 아메리카노 2300원, 카푸치노 2500원입니다.";  bool open = true;  char coffee[100];  int num, total = 0;  cout << menu << endl;  while (total < 20000){  cout << "주문>> ";  cin >> coffee >> num;  if (strcmp(coffee, "에스프레소") == 0) {  cout << 2000 \* num << "원입니다. 맛있게 드세요.\n";  total += 2000 \* num;  }  else if (strcmp(coffee, "아메리카노") == 0) {  cout << 2300 \* num << "원입니다. 맛있게 드세요.\n";  total += 2300 \* num;  }  else if (strcmp(coffee, "카푸치노") == 0) {  cout << 2500 \* num << "원입니다. 맛있게 드세요.\n";  total += 2500 \* num;  }  else {  cout << "다시 주문하세요!!\n" << endl;  }  }  cout << "오늘 " << total << "원을 판매하여 카페를 닫습니다. 내일 봐요 ~~~\n";  system("pause");  } |
| **[실행 결과]** |

1. 교재 94p 문제 16번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  int main() {  char buf[10000];  int total = 0, i = 0, arr[26] = { 0 };  cout << "영문 테스트를 입력하세요. 히스토그램을 그립니다.\n" << "택스트의 끝은 ; 입니다. 10000개까지 가능합니다.\n";  cin.getline(buf, 10000, ';');  while (buf[i] != '\0') {  if (isalpha(buf[i]) != 0) {  buf[i] = tolower(buf[i]);  total++;  int num = buf[i] - 97;  arr[num]++;  }  i++;  }  cout << "총 알파벳 수 " << total << endl;  for (i = 0; i < 26; i++) {  char alpha = 97 + i;  cout << alpha << " (" << arr[i] << " ) : ";  for (int j = 0; j < arr[i]; j++) {  cout << "\*";  }  cout << endl;  }  system("pause");  } |
| **[실행 결과]** |