|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_02 | 학번 : | 20135151 | 이름 : | 이갑성 |

* **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" //’’ “는 현재 폴더에서 소스프로그램에 있는곳에서 찾는다.  double Circle::getArea() { //Circlee 구현부  return 3.14\*radius\*radius;  }  **//Main.cpp**  #include <iostream> //<>는 설치되어있는 곳에서 찾는다  #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 //반지름 1000원 생성  Circle globalPizza(2000); //반지름 2000원 생성  void f() {  Circle fDonut(100); //지역객체  Circle fPizza(200);  }  int main() {  Circle mainDonut; //지역객체 //반지름 1원 생성  Circle mainPizza(30); //반지름 30원 생성  f(); //반지름 100원 생성, 반지름 200원 생성  } |
| **[실행결과]** |

* **내용 점검**

1. 교재 2장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | int main(){ return 0; } |
| 2 | 3번 |
| 3 | return 0; 라인 생략 가능 |
| 4 | 1. 오류발생 왜냐하면 C컴파일러는 절차지향이기 때문에 변수 초기화를 모두 맨위에서 이루어져야 합니다. 2. C++컴파일러는 C컴파일러와 달리 절차지향 언어가 아니기 때문이다. 3. 장점: 변수를 사용하기 직전에 선언함으로써 변수 이름에 대한 타이핑 오류를 줄일 수 있다.   단점: 선언된 변수를 일괄적으로 보기가 힘들고 코드사이에 있는 변수를 찾기 어렵다. |
| 5 | I love C++  I love programming |
| 6 | (1)using std::cout;  (2)using namespace std; |
| 7 | std::cin >> name; |
| 8 | using std::cout; |
| 9 | #include “myheader.h” |
| 10 | (1) O (2) O (3) O (4) X(char name[4]이여야 한다.) (5) X(공백문자 포함못함) |
| 11 | #include <cstring> |
| 12 | (1)Kitae님 환영합니다.  (2)Kitae님 환영합니다. |
| 13 | 4번 |
| 14 | 4번 |
| 15 | namespace |
| 16 | using namespace std |
| 17 | namespace std |
| 18 | #include <iostream> |
| 19 | cout << age << “ ” << pDept; |
| 20 | for(int m = 0; m <= n; m++){  cout << “\*”;  }  cout << endl; |

* 프로그램 과제

1. 교재 91p 문제 9번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  int main()  {  string name; //이름 객체 선언  string address; //주소 객체 선언  int age; //나이 객체 선언  cout << "이름은?";  getline(cin, name);  cout << "주소는?";  getline(cin, address);  cout << "나이는?";  cin >> age;  cout << name << ", " << address << ", " << age << "세";  return 0;  } |
| **[실행 결과]** |

1. 교재 91p 문제 10번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <cstring>  using namespace std;  int main(){  char str[10]; //문자열 객체 str선언  cout << "문자열 입력 >>";  cin.getline(str, 10);  for (int i = 0; i < strlen(str); i++)  {  for (int j = 0; j <= i; j++)  {  cout << str[j];  }  cout << endl;  }  return 0;  } |
| **[실행 결과]** |

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

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  int main()  {  int menu; //메뉴  int person; //몇인분  int pay; //주문가격    cout << "\*\*\*\*\* 승리장에 오신 것을 환영합니다. \*\*\*\*\* \n";  do {  cout << "========================== 메뉴 ========================== \n";  cout << "1.짬뽕(1500원)\n2.짜장(1000원)\n3.군만두(2000원)\n4.종료\n주문하시겠습니까?>> ";  cin >> menu;  if (menu == 4)  break;  else if (menu >= 5)  {  cout << "다시 주문하세요!!" << endl;  continue;  }  cout << "몇인분?";  cin >> person;  if (menu == 1)  {  pay = 1500 \* person;  cout << "짬뽕 " << person << "인분 나왔습니다. 총 금액은 " << pay << "원 입니다. \n";  }  else if (menu == 2)  {  pay = 1000 \* person;  cout << "짜장 " << person << "인분 나왔습니다. 총 금액은 " << pay << "원 입니다. \n";  }  else if (menu == 3)  {  pay = 2000 \* person;  cout << "군만두 " << person << "인분 나왔습니다. 총 금액은 " << pay << "원 입니다. \n";  }  } while (1);  cout << "오늘 영업은 끝났습니다. \n";  return 0;  } |
| **[실행 결과]** |

1. 교재 92p 문제 14번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  int main()  {  int total = 0; //총 금액  int pay; //지불금액  string coffee; //커피이름을 저장할 배열  int many; //주문 수량  cout << "에스프레소 2000원, 아메리카노 2300원, 카푸치노 2500원입니다." << endl;  while (total < 20000)  {  cout << "주문>>";  cin >> coffee >> many;  if (coffee == "에스프레소")  {  pay = 2000 \* many;  cout << pay << "원입니다. 맛있게 드세요" << endl;  total = total + pay;  }  else if (coffee == "아메리카노")  {  pay = 2300 \* many;  cout << pay << "원입니다. 맛있게 드세요" << endl;  total = total + pay;  }  else if (coffee == "카푸치노")  {  pay = 2500 \* many;  cout << pay << "원입니다. 맛있게 드세요 \n";  total = total + pay;  }  }  cout << "오늘 " << total << "원을 판매하여 카페를 닫습니다. 내일 봐요~~~~~";  return 0;  } |
| **[실행 결과]** |

1. 교재 94p 문제 16번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <cstring>  using namespace std;  int main()  {  char text[100001]; //10000개까지 가능하게 하기위해서  int cnt = 0; //알파벳의 개수  int alpha[26] = { 0 };  cout << "영문 텍스트를 입력하세요. 히스토그램을 그립니다. \n";  cout << "텍스트의 끝은 ; 입니다. 10000개까지 가능합니다. \n";  cin.getline(text, 100001, ';');    for (int i = 0; i < strlen(text); i++) //총 알파벳 수 찾는 for문  {  if (isalpha(text[i]) != 0)  {  cnt++;  }  }  for (int i = 0; i < strlen(text); i++)  {  if (tolower(text[i]) == 'a')  {  alpha[0]++;  }  else if (tolower(text[i]) == 'b')  {  alpha[1]++;  }  else if (tolower(text[i]) == 'c')  {  alpha[2]++;  }  else if (tolower(text[i]) == 'd')  {  alpha[3]++;  }  else if (tolower(text[i]) == 'e')  {  alpha[4]++;  }  else if (tolower(text[i]) == 'f')  {  alpha[5]++;  }  else if (tolower(text[i]) == 'g')  {  alpha[6]++;  }  else if (tolower(text[i]) == 'h')  {  alpha[7]++;  }  else if (tolower(text[i]) == 'i')  {  alpha[8]++;  }  else if (tolower(text[i]) == 'j')  {  alpha[9]++;  }  else if (tolower(text[i]) == 'k')  {  alpha[10]++;  }  else if (tolower(text[i]) == 'l')  {  alpha[11]++;  }  else if (tolower(text[i]) == 'm')  {  alpha[12]++;  }  else if (tolower(text[i]) == 'n')  {  alpha[13]++;  }  else if (tolower(text[i]) == 'o')  {  alpha[14]++;  }  else if (tolower(text[i]) == 'p')  {  alpha[15]++;  }  else if (tolower(text[i]) == 'q')  {  alpha[16]++;  }  else if (tolower(text[i]) == 'r')  {  alpha[17]++;  }  else if (tolower(text[i]) == 's')  {  alpha[18]++;  }  else if (tolower(text[i]) == 't')  {  alpha[19]++;  }  else if (tolower(text[i]) == 'u')  {  alpha[20]++;  }  else if (tolower(text[i]) == 'v')  {  alpha[21]++;  }  else if (tolower(text[i]) == 'w')  {  alpha[22]++;  }  else if (tolower(text[i]) == 'x')  {  alpha[23]++;  }  else if (tolower(text[i]) == 'y')  {  alpha[24]++;  }  else if (tolower(text[i]) == 'z')  {  alpha[25]++;  }  }  cout << "총 알파벳 수 " << cnt << endl;  for (int i = 0; i < 23; i++)  {  cout << (char)(97 + i) << "(" << alpha[i] << ") : ";  for (int j = 0; j < alpha[i]; j++)  {  cout << "\*";  }  cout << endl;  }  return 0;  } |
| **[실행 결과]** |