|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_12 | 학번 : | 20145128 | 이름 : | 박인근 |

* **강의 내용**

|  |
| --- |
| 1. 제네릭 함수 |
| #include <iostream>  using namespace std;  template <class T>  void print(T array [], int n) {  for(int i=0; i<n; i++)  cout << array[i] << '\t';  cout << endl;  }  void print(char array [], int n) { // 템플릿 함수와 동일한 이름의 함수 중복  for(int i=0; i<n; i++)  cout << (int)array[i] << '\t'; // array[i]를 int 타입으로 변환하여 정수 출력  cout << endl;  }  int main() {  int x[] = {1,2,3,4,5};  double d[5] = { 1.1, 2.2, 3.3, 4.4, 5.5 };  print(x, 5); // 템플릿으로부터 구체화한 함수 호출  print(d, 5); // 템플릿으로부터 구체화한 함수 호출  char c[5] = {1,2,3,4,5};  print(c, 5); // char 배열을 숫자로 출력하는 중복 함수 호출  } |
| **[실행결과]** |

|  |
| --- |
| 1. 제네릭 클래스 |
| #include <iostream>  using namespace std;  template <class T1, class T2> // 두 개의 제네릭 타입 선언  class GClass {  T1 data1;  T2 data2;  public:  GClass();  void set(T1 a, T2 b);  void get(T1 &a, T2 &b);  };  template <class T1, class T2>  GClass<T1, T2>::GClass() {  data1 = 0; data2 = 0;  }  template <class T1, class T2>  void GClass<T1, T2>::set(T1 a, T2 b) {  data1 = a; data2 = b;  }  template <class T1, class T2>  void GClass<T1, T2>::get(T1 & a, T2 & b) {  a = data1; b = data2;  }  int main() {  int a;  double b;  GClass<int, double> x;  x.set(2, 0.5); // int, double의 매개 변수  x.get(a, b);  cout << "a=" << a << '\t' << "b=" << b << endl;  char c;  float d;  GClass<char, float> y;  y.set('m', 12.5); // char, double의 매개 변수  y.get(c, d);  cout << "c=" << c << '\t' << "d=" << d << endl;  } |
| **[실행결과]** |

|  |
| --- |
| 1. STL(Standard Template Library 활용 - vector 컨테이너 |
| #include <iostream>  #include <string>  #include <vector>  using namespace std;  int main() {  vector<string> sv; // 문자열 벡터 생성  string name;  cout << "이름을 5개 입력하라" << endl;  for (int i = 0; i<5; i++) { // 한 줄에 한 개씩 5 개의 이름을 입력받는다.  cout << i + 1 << ">>";  getline(cin, name);  sv.push\_back(name);  }  name = sv.at(0); // 벡터의 첫 번째 이름  for (int i = 1; i<sv.size(); i++) {  if (name < sv[i]) // sv[i]의 문자열이 name보다 사전에서 뒤에 나옴  name = sv[i]; // name을 sv[i]의 문자열로 변경  }  cout << "사전에서 가장 뒤에 나오는 이름은 " << name << endl;  } |
| **[실행결과]** |

* 내용 점검 – 교재9장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 다형성 |
| 5 | 1. 기본 클래스 : Base 파생 클래스 : Derived 2. Derived::f() called 3. Base::f() called 4. Base::f() called 5. Base::f() called |
| 6 | 1. 1,2,3 2. C::f() called 3. C::f() called 4. C::f() called 5. C::f() called |
| 7 | 가까운, 범위 규칙, 범위 지정 연산자 |
| 8 | 1. ::f(); 2. A::f(); 3. f(); |
| 9 | 2 |
| 10 | 1. id=10, 문제점은 ~Student()가 실행되지 않음 2. ~Person() { cout << “id=” << id << endl; }  * Virtual ~Person() { cout << “id=” << id << endl; } |
| 11 | 3 |
| 12 | 3 |
| 13 | 2 |
| 14 | (1) 1,3  (2) Circle 클래스에  void draw() { cout << “반지름=” << radius << “인 원” << endl;  을 추가합니다. |
| 15 | 4 |

* 프로그램 과제

|  |
| --- |
| 1. 교재 470p : 문제 9번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Printer {  protected:  string Mname; // 모델명  string manufac; // 제조사  int pCount; // 인쇄 매수  int avaCount; // 인쇄 종이 잔량  public:  Printer(string Mname, string manufac, int ac) {  this->Mname = Mname;  this->manufac = manufac;  this->avaCount = ac;  }  virtual void print(int pages) = 0;  virtual void show() = 0;  string getMname() { return Mname; }  string getManufac() { return manufac; }  int getavaCount() { return avaCount; }  };  class InkjetPrinter : public Printer {  int avaInk; // 잉크 잔량  public:  InkjetPrinter(int avaInk) : Printer("Officejet V40", "HP", 5) {  this->avaInk = avaInk;  }  void print(int pages) {  pCount = pages;  if (avaCount < pages) cout << "용지가 부족하여 프린트 할 수 없습니다." << endl;  else {  cout << "프린트하였습니다." << endl;  avaCount -= pCount;  avaInk -= pCount;  }  }  void show() {  cout << getMname() << " ," << getManufac() << " ,남은 종이 " << getavaCount() << "장 ,남은 잉크 " << avaInk << endl;  }  };  class LaserPrinter : public Printer {  int avaToner;  public:  LaserPrinter(int avaToner) : Printer("SCX-6x45", "삼성전자", 3) {  this->avaToner = avaToner;  }  void print(int pages) {  pCount = pages;  if (avaCount < pages) cout << "용지가 부족하여 프린트 할 수 없습니다." << endl;  else {  cout << "프린트하였습니다." << endl;  avaCount -= pCount;  avaToner -= pCount;  }  }  void show() {  cout << getMname() << " ," << getManufac() << " ,남은 종이 " << getavaCount() << "장 ,남은 토너 " << avaToner << endl;  }  };  int main() {  InkjetPrinter \*p = new InkjetPrinter(10);  LaserPrinter \*q = new LaserPrinter(20);  int print, cnt;  char answer;  cout << "현재 작동중인 2 대의 프린터는 아래와 같다" << endl;  cout << "잉크젯 : "; p->show();  cout << "레이저 : "; q->show();  while (1) {  cout << "프린터(1:잉크젯, 2:레이저)와 매수 입력>>";  cin >> print >> cnt;  if (print == 1) {  p->print(cnt);  p->show();  q->show();  }  if (print == 2) {  q->print(cnt);  p->show();  q->show();  }  cout << "계속 프린트 하시겠습니까(y/n)>>";  cin >> answer;  if (answer == 'y') continue;  else break;  }  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 525p : 문제2번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  template <class T>  bool equalArrays(T arr1[], T arr2[], int n) {  for (int i = 0; i < n; i++) {  if (arr1[i] != arr2[i]) return false;  }  return true;  }  int main() {  int x[] = { 1, 10, 100, 5, 4 };  int y[] = { 1, 10, 100, 5, 4 };  int z[] = { 1, 10, 15, 25, 30 };    if (equalArrays(x, y, 5)) cout << "같다" << endl; //배열 x,y 가 같으므로 "같다" 출력  else cout << "다르다" << endl;  if (equalArrays(x, z, 5)) cout << "같다" << endl; //배열 x,z 가 다르므로 "다르다" 출력  else cout << "다르다" << endl;  if (equalArrays(x, z, 2)) cout << "같다" << endl; //배열 x,z 의 0,1 번째 원소는 같으므로 "같다" 출력  else cout << "다르다" << endl;  if (equalArrays(x, z, 3)) cout << "같다" << endl; //배열 x,z 의 2 번째 원소부터는 다르므로 "다르다" 출력  else cout << "다르다" << endl;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 525p : 문제4번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  template <class T>  bool search(int s, T arr[], int size) {  for (int i = 0; i < size; i++) {  if (arr[i] == s) return true;  }  return false;  }  int main() {  int x[] = { 1, 10, 100, 5, 4 };  if (search(100, x, 5)) cout << "100이 배열 x에 포함되어 있다" << endl;  else cout << "100이 배열 x에 포함되어 있지 않다" << endl;  if (search(7, x, 5)) cout << "7이 배열 x에 포함되어 있다" << endl;  else cout << "7이 배열 x에 포함되어 있지 않다" << endl;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 526p : 문제6번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  template <class T>  T\* remove(T src[], int sizeSrc, T minus[], int sizeMinus, int& retSize) {  T\* newArr = new T;  int a = sizeMinus;  int cnt = 0;  for (int i = 0; i < sizeSrc; i++) {  for (int j = 0; j < sizeMinus; j++) {  if (src[i] != minus[j]) a--; // src[i]번째 원소와 minus[] 배열의 전체 원소를 비교하며 확인  }  if (a == 0) { // src[i] 번째 원소가 minus[] 배열에 없을 경우  newArr[cnt] = src[i]; // 새로운 배열에 src[i]번째 원소 추가  cnt++;  }  a = sizeMinus; // a를 다시 초기화  }  retSize = cnt; //배열의 크기를 전달받음  return newArr;  }  int main() {  int x[] = { 1,10,100,5,4 };  int y[] = { 1000, 100, 10, 1 };  int z[] = { 5, 10, 15, 20 };  int size = 0; // 배열 크기를 저장하기 위한 변수  int\* a = remove(x, 5, y, 4, size);  cout << "x,y에 대한 새로운 배열 = { ";  for (int i = 0; i < size; i++) {  cout << a[i] << ' ';  }  cout << "}" << endl;    cout << "x,z에 대한 새로운 배열 = { ";  int\* b = remove(x, 5, z, 4, size);  for (int i = 0; i < size; i++) {  cout << b[i] << ' ';  }  cout << "}" << endl;  system("pause");  } |
| **[실행결과]** |