|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_12 | 학번 : | 20145165 | 이름 : | 정균모 |

* **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)base, derived (2)derived f (3)base f (4)base f (5)base f |
| 6 | (1)1,2,3 (2)c::f (3)c::F (4)c::F (5)c::f |
| 7 | 가까운, 범위 규칙, 범위 지정 연산자 |
| 8 | 1. F:: (2) A::F (3) F:: |
| 9 | 2 |
| 10 | (1) 상속을 받을 때 public 을 선언 해주지 않아서 main에서 업 캐스팅이 안된다.  (2) class Student : public Person |
| 11 | 3 |
| 12 | 3 |
| 13 | 2 |
| 14 | (1)1, 3  (2) #include <iostream>  #include <string>  #include <vector>  using namespace std;  class Shape {  public:  void paint() { draw(); }  virtual void draw() = 0;  };  class Circle : public Shape {  int radius;  public:  Circle(int radius = 1) { this->radius = radius; }  double getArea() { return 3.14\*radius\*radius; }  void draw() { cout << "반지름 =" << radius << "인 원" << endl; }  };  int main() {  Circle \*p = new Circle(10);  p->paint();  system("pause");  } |
| 15 | 4 |

* 프로그램 과제

|  |
| --- |
| 1. 교재 470p : 문제 9번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Printer {  string model;  string manufacturer;  int printedCount;  int availableCount;  protected:  Printer(string model, string manufacturer, int pages) {  this->model = model; this->manufacturer = manufacturer; this->availableCount = pages; printedCount = 0;  }  void print(int pages) {  printedCount += pages;  availableCount -= pages;  }  string Model() { return model; }  string Manufacturer() { return manufacturer; }  int PrintedCount() { return printedCount; }  int AvailableConut() { return availableCount; }  };  class InkJetPrinter : virtual protected Printer {  int availableInk;  public:  InkJetPrinter(string model, string manufacturer, int pages, int ink) : Printer(model, manufacturer, pages) {  availableInk = ink;  }  int printInkJet(int pages) {  int sw;  if (pages <= AvailableConut() && pages <= availableInk) {  print(pages);  availableInk -= pages;  sw = 1;  }  else if (pages > AvailableConut() && pages <= availableInk)  sw = 2;  else if (pages <= AvailableConut() && pages > availableInk)  sw = 3;  else  sw = 4;  return sw;  }  string getModel() { return Model(); }  string getManufacturer() { return Manufacturer(); }  int getAvailableConut() { return AvailableConut(); }  int getAvailableInk() { return availableInk; }  };  class LaserPrinter :virtual protected Printer {  int availableToner;  public:  LaserPrinter(string model, string manufacturer, int pages, int toner) :Printer(model, manufacturer, pages) {  availableToner = toner;  }  int printLaser(int pages) {  int sw;  if (pages <= AvailableConut() && pages <= availableToner) {  print(pages);  availableToner -= pages;  sw = 1;  }  else if (pages > AvailableConut() && pages <= availableToner)  sw = 2;  else if (pages <= AvailableConut() && pages > availableToner)  sw = 3;  else  sw = 4;  return sw;  }  string getModel() { return Model(); }  string getManufacturer() { return Manufacturer(); }  int getAvailableConut() { return AvailableConut(); }  int getAvailableToner() { return availableToner; }  };  int main() {  InkJetPrinter \*ip = new InkJetPrinter("Officejet V40", "HP", 5, 10);  LaserPrinter \*lp = new LaserPrinter("SCX-6x45", "삼성전자", 3, 20);  cout << "현재 작동중인 2 대의 프린트는 아래와 같다" << endl;  cout << "잉크젯 : " << ip->getModel() << " ," << ip->getManufacturer();  cout << " ,남은 종이" << ip->getAvailableConut();  cout << "장 ,남은 잉크 " << ip->getAvailableInk() << endl;  cout << "레이저 : " << lp->getModel() << " ," << lp->getManufacturer();  cout << " ,남은 종이" << lp->getAvailableConut();  cout << "장 ,남은 잉크 " << lp->getAvailableToner() << endl << endl;  for (;;) {  cout << "프린터(1:잉크젯, 2:레이저)와 매수 입력>>";  int sw, n;  cin >> sw >> n;  if (sw == 1) {  switch (ip->printInkJet(n)) {  case 1:  cout << "프린트하였습니다" << endl;  break;  case 2:  cout << "용지가 부족하여 프린트할 수 없습니다." << endl;  break;  case 3:  cout << "잉크가 부족하여 프린트할 수 없습니다." << endl;  break;  case 4:  cout << "용지와 잉크가 모두 부족하여 프린트할 수 없습니다." << endl;  break;  }  cout << "잉크젯 : " << ip->getModel() << " ," << ip->getManufacturer();  cout << " ,남은 종이" << ip->getAvailableConut();  cout << "장 ,남은 잉크 " << ip->getAvailableInk() << endl;  cout << "레이저 : " << lp->getModel() << " ," << lp->getManufacturer();  cout << " ,남은 종이" << lp->getAvailableConut();  cout << "장 ,남은 잉크 " << lp->getAvailableToner() << endl;  }  else if (sw == 2) {  switch (lp->printLaser(n)) {  case 1:  cout << "프린트하였습니다" << endl;  break;  case 2:  cout << "용지가 부족하여 프린트할 수 없습니다." << endl;  break;  case 3:  cout << "토너가 부족하여 프린트할 수 없습니다." << endl;  break;  case 4:  cout << "용지와 토너가 모두 부족하여 프린트할 수 없습니다." << endl;  break;  }  cout << "잉크젯 : " << ip->getModel() << " ," << ip->getManufacturer();  cout << " ,남은 종이" << ip->getAvailableConut();  cout << "장 ,남은 잉크 " << ip->getAvailableInk() << endl;  cout << "레이저 : " << lp->getModel() << " ," << lp->getManufacturer();  cout << " ,남은 종이" << lp->getAvailableConut();  cout << "장 ,남은 잉크 " << lp->getAvailableToner() << endl;  }  else cout << "프린트 종류 입력 에러" << endl;  cout << "계속 프린트 하시겠습니까(y/n)>>";  char a;  cin >> a;  if (a == 'n') break;  cout << endl << endl;  }  delete ip; delete lp;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 525p : 문제2번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  template <class T>  bool equalArrays(T x[], T y[], int n) { // 배열이 같은지 확인하는 함수  for (int i = 0; i < n; ++i) {  if (x[i] != y[i])  return false;  }  return true;  }  int main() {  int x[] = { 1,10,100,5,4 }; // 비교할 두 int 배열  int y[] = { 1,10,100,5,4 };  char a[] = { 'a','d','z','t','w' }; // 비교할 두 char 배열  char b[] = { 'k','a','d','z','t' };  if (equalArrays(x, y, 5))cout << "같다" << endl; // int 배열 비교  else cout << "다르다" << endl;;  if (equalArrays(a, b, 5))cout << "char 배열 같다" << endl; // char 배열 비교  else cout << "다르다" << endl;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 525p : 문제4번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  template <class T>  bool search(T data, T x[], int n) {  for (int i = 0; i < n; ++i) {  if (data == x[i]) 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; // 이 cout 실행  else cout << "100이 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) {  bool sw;  T \*p = new T[sizeSrc];  for (int i = 0; i < sizeSrc; ++i) {  sw = true;  for (int j = 0; j < sizeMinus; ++j) {  if (src[i] == minus[j]) {  sw = false;  break;  }  }  if (sw) {  p[retsize] = src[i];  ++retsize;  }  }  return p;  }  int main() {  int a[] = { 1,2,3,4,5,6,7,8,9,10 };  int b[] = { 5,6,7,8,9 };  int size = 0;  int \*p = remove(a, 10, b, 5, size);  for (int i = 0; i < size; ++i)  cout << p[i] << ' ';  cout << endl;  delete[] p;  size = 0;  char c[] = { 'a','b','i','m','c','d','e', };  char d[] = { 'k','i','m','n','u' };  char \*q = remove(c, 7, d, 5, size);  for (int i = 0; i < size; ++i)  cout << q[i] << ' ';  cout << endl;  delete[] q;  system("pause");  } |
| **[실행결과]** |