|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_12 | 학번 : | 20135151 | 이름 : | 이갑성 |

* **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 | 1번 |
| 2 | 2번 |
| 3 | 3번 |
| 4 | 다형성 |
| 5 | (1) 기본클래스: Base 파생클래스: Derrived |
| 6 | (1) 1번 |
| 7 | 범위 지정 연산자. |
| 8 | (1) ::f();  (2) A::f();  (3) f(); |
| 9 | 2번 |
| 10 | (1)  실행 결과    Student 메모리를 할당했는데 Student의 메모리를 소멸시켜주지 않았다.  (2)  #include <iostream>  using namespace std;  class Person {  int id;  public:  Person(int id = 0) {  this->id = id;  }  virtual ~Person() {  cout << "id = " << id << endl;  }  };  class Student :public Person {  char \*name;  public:  Student(int id, const char\* name) : Person(id) {  int len = strlen(name);  this->name = new char[len + 1];  strcpy(this->name, name);  }  virtual ~Student() {  cout << "name = " << name << endl;  delete[] name;  }  };  int main() {  Person \*p = new Student(10, "손연재");    delete p;  } |
| 11 | 3번 |
| 12 | 3번 |
| 13 | 2번 |
| 14 | (1) 1 ,2 ,3 ,4 번  (2)  void draw(){  cout << “반지름=” << this->radius << “인 원” << endl;  } |
| 15 | 4번 |

* 프로그램 과제

|  |
| --- |
| 1. 교재 470p : 문제 9번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Printer {  protected:  string model; //프린트 모델명  string manufactuerer; //제조사  int printedCount; //인쇄 매수  int availableCount; //인쇄 종이 잔량  public:  Printer(string model, string manufactuerer, int availableCount)  {  this->model = model;  this->manufactuerer = manufactuerer;  this->availableCount = availableCount;  }  virtual void print(int pages) = 0; //page만큼 프린트하는 함수  void show(); //현재 프린트의 모델, 제조사, 인쇄 매수, 인쇄 종이 잔량 등을 출력하는 함수  };  void Printer::show() {  cout << model << ", " << manufactuerer << ", 남은 종이 " << availableCount << "장, ";  }  class InkPrinter : public Printer {  int availableInk; //잉크 잔량  public:  InkPrinter(string model, string manufactuerer, int availableCount, int availableInk) : Printer(model, manufactuerer, availableCount){  this->availableInk = availableInk;  }  void show()  {  Printer::show();  cout << "남은잉크 " << this->availableInk << endl;  }  void print(int pages) {  int tmp1, tmp2;  tmp1 = availableCount;  tmp2 = availableInk;  availableCount -= pages;  availableInk -= pages;  if (availableCount < 0)  {  cout << "용지가 부족하여 프린트 할 수 없습니다." << endl;  availableCount = tmp1;  availableInk = tmp2;  }  else if (availableInk < 0)  {  cout << "잉크가 부족하여 프린트 할 수 없습니다." << endl;  availableCount = tmp1;  availableInk = tmp2;  }  else  cout << "프린트하였습니다." << endl;  }  };  class LaserPrinter : public Printer {  int availableToner; //토너 잔량  public:  LaserPrinter(string model, string manufactuerer, int availableCount, int availableToner) : Printer(model, manufactuerer, availableCount) {  this->availableToner = availableToner;  }  void show()  {  Printer::show();  cout << "남은토너 " << this->availableToner << endl;  }  void print(int pages)  {  int tmp1, tmp2;  tmp1 = availableCount;  tmp2 = availableToner;  availableCount -= pages;  availableToner -= pages;  if (availableCount < 0)  {  cout << "용지가 부족하여 프린트 할 수 없습니다." << endl;  availableCount = tmp1;  availableToner = tmp2;  }  else if (availableToner < 0)  {  cout << "토너가 부족하여 프린트 할 수 없습니다." << endl;  availableCount = tmp1;  availableToner = tmp2;  }  else  {  cout << "프린트하였습니다." << endl;  }  }  };  int main()  {  cout << "현재 작동중인 2 대의 프린터는 아래와 같다." << endl;  InkPrinter ink("Officejet V40", "HP", 5, 10);  LaserPrinter laser("SCX-6X45", "삼성전자", 3, 20);  char more;  int menu, cnt;  cout << "잉크젯: ";  ink.show();  cout << "레이저: ";  laser.show();  do {  cout << "프린터(1:잉크젯, 2:레이저)와 매수 입력 >> ";  cin >> menu >> cnt;  if (menu == 1)  {  ink.print(cnt);  }  else if (menu == 2)  {  laser.print(cnt);  }  ink.show();  laser.show();  cout << "계속 프린트 하시겠습니까?(y/n) >> ";  cin >> more;  cout << endl << endl;  } while (more != 'n');  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 525p : 문제2번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  template <class T>  T biggest(T array[], int n)  {  T biggest = 0;  for (int i = 0; i < n; i++)  {  if (biggest < array[i])  {  biggest = array[i];  }  }  return biggest;  }  int main()  {    int x[] = { 1, 2, 3, 4, 5 };  double d[] = { 10.5, 20.1, 11.1, 3.7 };  for (int i = 0; i < 5; i++)  {  cout << x[i] << ' ';  }  cout << endl << "위의 값들 중에 가장 큰 값: " << biggest(x, 5) << endl;  for (int i = 0; i < 4; i++)  {  cout << d[i] << ' ';  }  cout << endl << "위의 값들 중에 가장 큰 값: " << biggest(d, 4) << endl;;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 525p : 문제4번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  template <class T>  bool search(T findout, T array[], int n)  {  bool flag = false;  for (int i = 0; i < n; i++)  {  if (array[i] == findout)  {  flag = true;  }  }  return flag;  }  int main()  {  char find;  char x[] = {'c', 'a', 'e', 'z', 'f'};  cout << "배열 x에 찾고자하는 원소를 찾으세요 >> ";  cin >> find;  if (search(find, x, 5))  cout << find << "가 배열 x에 포함되어 있다." << endl;  else  cout << find << "가 배열 x에 포함되어 있지 않다." << endl;  } |
| **[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)  {  int index = 0;  bool flag;  T \*p;  for (int i = 0; i < sizeSrc; i++)  {  flag = false;  for (int j = 0; j < sizeMinus; j++)  {  if (src[i] == minus[j])  flag = true;  }  if (flag == false)  {  index++;  }  }  p = new T[index];  retSize = index;  for (int i = 0; i < sizeSrc; i++)  {  flag = false;  for (int j = 0; j < sizeMinus; j++)  {  if (src[i] == minus[j])  flag = true;  }  if (flag == false)  p[i] = src[i];  }  return p;  delete[] p;  }  int main()  {  int x[] = { 'a', 'b', 'c' };  int y[] = { 'b' ,'c' ,'d' };  int sizeResult;  int \*result = remove(x, 3, y, 3, sizeResult);  for (int i = 0; i < sizeResult; i++)  {  cout << (char)result[i] << ' ';  }  } |
| **[실행결과]** |