|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2018\_2\_C++ \_12 | 학번 : | 20157135 | 이름 : | 이상우 |

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  } |
| **[실행 결과]** |

* 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;  } |
| **[실행 결과]** |

* 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;  } |
| **[실행 결과]** |

* STL(Standard Template Library 활용 - iterator 사용

|  |
| --- |
| #include <iostream>  #include <vector>  using namespace std;  int main() {  vector<int> v; //정수 벡터 생성  v.push\_back(1);  v.push\_back(2);  v.push\_back(3);  vector<int>::iterator it; // 벡터 v의 원소에 대한 포인터 it 선언  for (it = v.begin(); it != v.end(); it++) { // iterator를 이용하여 모든 원소 탐색  int n = \*it; // it가 가리키는 원소 값 리턴  n = n \* 2; // 곱하기 2  \*it = n; // it가 가리키는 원소에 값 쓰기  }  for (it = v.begin(); it != v.end(); it++) // 벡터 v의 모든 원소 출력  cout << \*it << ' ';  cout << endl;  } |
| **[실행 결과]** |

* 내용 점검

1. 교재 10장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | 1. 템플릿을 선언하기 위해 사용하는 키워드는 template 이나 generic 이다 |
| 2 | 1. 템플릿을 사용하면 컴파일 오류 메시지가 풍부하여 디버깅에 많은 도움을 준다. |
| 3 | 1. Template (class T) |
| 4 | 2) 컴파일러 |
| 5 | template <class T>  Bool equal(int a, int b){  if(a == b){ return true; }else{ return false;} } |
| 6 | template <class T>  Void insert(int a, int b[ ], int index){  b[index] = a; } |
| 7 | Template <class T>  T max(T x, T y){  If(x > y)  return x;  else  return y;  } |
| 8 | template <class T>  Bool equals(TYPE x, TYPE y){  if(x == y)  return true;  else  return false;  } |
| 9 | 1) int avg(int \*p, int n){  int k;  int sum = 0;  for(k = 0; k < n; k++)  sum += p[k];  return sum/n;  }  2) double avg(double \*p, int n){  int k;  double sum = 0;  for(k = 0; k < n; k++)  sum += p[k];  return sum/n;  } |
| 10 | 1) 템플릿 함수와 보통 함수는 중복하여 공존 가능  2) 3.14  3)special 100 |
| 11 | 3. 탬플릿 함수와 동일한 이름의 함수가 중복되어 있을 때 중복 함수우선적으로 바이 |
| 12 | 1. T\* p; // T 타입의 포인터 p를 선언하라.   int size; // 배열에 현재 들어 있는 개수를 나타내는 변수 size를 선언하라.   1. Template<class T> Container<T> :: Container(int n){   P = new T[n];  Size = n;  }   1. template<class T> Container<T> :: ~Container(){   Delete [] p;  }   1. template<class T>Container<T> :: get(int index){   return p[index];  }   1. int main(){   Container<char> c(26);  }   1. #include <iostream>   using namespace std;  int main(){  Container<char> c(26); |

* 프로그램 과제

1. 교재 525p : 문제 2번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  template <class T>  bool equalArrays(T a[], T b[], int size) {  for (int i = 0; i < size; i++)  if (a[i] != b[i]) return false;  return true;  }  int main() {  int x[] = { 1, 10, 100, 5, 4 };// x 배열  int y[] = { 1, 10, 100, 5, 4 };// y 배열  if (equalArrays(x, y, 5)) cout << "같다" << endl;  else cout << "다르다" << endl;  char ch;  ch = getchar();  } |
| **[실행 결과]** |

1. 교재 525p : 문제 3번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  template <class T>  void reverseArray(T a[], int size) {  T imsi;  for (int i = 0; i < size / 2; i++) {  imsi = a[i];  a[i] = a[size - i - 1];  a[size - i - 1] = imsi;  }  }  int main() {  int x[] = { 1, 10, 100, 5, 4 };  reverseArray(x, 5);  for (int i = 0; i < 5; i++) cout << x[i] << ' ';  char ch;  ch = getchar();  } |
| **[실행 결과]** |

1. 교재 525p : 문제 4번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  template <class T>  bool search(T se, T a[], int size) {  for (int i = 0; i < size; i++)  if (a[i] == se) 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;  char ch;  ch = getchar();  } |
| **[실행 결과]** |

1. 교재 527p : 문제 8번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  class Circle;  class Comparable {  public:  virtual bool operator > (Circle& op2) = 0; // 연산자 >  virtual bool operator < (Circle& op2) = 0; // 연산자 <  virtual bool operator == (Circle& op2) = 0; // 연산자 ==  };  class Circle : public Comparable {  int radius;  public:  Circle(int radius = 1) { this->radius = radius; }  int getRadius() { return radius; }  bool operator > (Circle& op2) {  if (this->radius > op2.radius) return true;  else return false;  }  bool operator < (Circle& op2) {  if (this->radius < op2.radius) return true;  else return false;  }  bool operator == (Circle& op2) {  if (this->radius == op2.radius) return true;  else return false;  }  };  template <class T>  T bigger(T a, T b) {  if (a > b) return a;  else return b;  }  int main() {  int a = 20, b = 50, c;  c = bigger(a, b);  cout << "20과 50중 큰 값은 " << c << endl;  Circle waffle(10), pizza(20), y;  y = bigger(waffle, pizza);  cout << "waffle과 pizza 중 큰 것의 반지름은 " << y.getRadius() << endl;  char ch;  ch = getchar();  } |
| **[실행 결과]** |

1. 교재 527p : 문제 9번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <vector> // vector 튜토리얼  using namespace std;  int main() {  vector<int> v; // 템플릿 선언  int n;  double avg;  while (1) {  cout << "정수를 입력하세요(0을 입력하면 종료) >>";  cin >> n;  v.push\_back(n);  if (n == 0)  break;  int sum = 0;  for (int i = 0; i < v.size(); i++) {  cout << v.at(i) << " ";  sum += v.at(i);  }  avg = (double)sum / v.size();  cout << endl << "평균 = " << avg << endl;  char ch;  ch = getchar();  ch = getchar();  }  } |
| **[실행 결과]** |