|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cpp\_15주 | 학번 : | 20175260 | 이름 : | 최영빈 |

* **개념 확인 – 템플릿과 STL**

1. 다음 질문에 대하여 O, X 하시오.
2. 템플릿을 선언하기 위해 사용하는 키워드는 template이나 generic이다 ( x )
3. 제네릭 타입을 선언하는 키워드는 class 이다 ( o )
4. 템플릿 함수와 동일한 이름의 함수가 중복되어 있을 경우 템플릿 함수가 우선적으로 바인딩 된다( x )
5. C++ 표준STL 라이브러리가 작성된 이름공간은 std 이다 ( o )
6. 제네릭 함수나 제네릭 클래스를 활용하는 프로그래밍 기법을 일반화 프로그래밍이라고 한다 ( o )
7. 템플릿 함수는 오버로딩을 할 수 없다( x )
8. 템플릿을 사용하면 컴파일 오류메시지가 풍부하여 디버깅에 많은 도움을 준다 ( x )
9. map 컨테이너 요소들은 키값을 기준으로 내림차순으로 정렬된다 ( x )
10. 람다식에서 리턴타입은 생략할 수 있다 ( o )
11. 람다식에서 함수 바깥의 변수 목록을 사용하려면 캡쳐리스트로 작성하면 된다 ( o )
12. 다음 STL의 각 기능을 사용하기 위해 필요한 헤더 파일을 제시하시오
13. vector 클래스 <vector>
14. merge 함수 <algorithm>
15. map 클래스 <map>
16. 다음에서 템플릿 선언을 잘못한 것은? 2
17. template <class T>
18. template (class T) <————————
19. template <typename T>
20. template <typename T1, typename T2>
21. 다음 제네릭 함수 선언에서 잘못된 부분을 바르게 수정하시오

template <typename T>

int max(T x, T, y){

if(x>y) return x;

else return y;

}

===== 풀이 =====

자료형을 int -> T로 변경

1. 다음 두 함수를 일반화한 제네릭 함수를 작성하시오

void insert(int a, int \*b, int index){

b[index] = a;

}

void insert(char a, char \*b, int index){

\*(b+index) = a;

}

===== 풀이 =====

template <typename T>

void insert(T a,T \*b, int index){  
 \*(b+index) = a;

}

1. 제시된 문제를 해결하는 문장을 제시하시오

vector<double> v;

1. 벡터 v에 마지막 원소로 3.1 삽입

===== 풀이 =====

v.push\_back(3.1);

1. 벡터 v에 저장된 원소 개수

===== 풀이 =====

v.size();

1. 벡터 v에 저장된 첫번째 원소 삭제

===== 풀이 =====

v.erase(v.begin());

1. vector<char> v; 벡터 v에 저장된 모든 원소를 제시된 방법을 사용하여 출력하는 문장을 작성하시오.
2. 반복자 사용

===== 풀이 =====

vector<char>::iterator it;

for(it = v.begin(); it != v.end(); it++)

cout<<\*it<<endl;

1. 범위기반 for문 사용

===== 풀이 =====

for(auto &vv: v)

cout<<vv<<endl;

1. 매개변수로 배열 크기와 double 타입의 배열을 받아 오름차순으로 정렬하는 람다식을 auto 변수 method에 저장하고 호출하는 문장을 작성하시오. 단, 오름차순 정렬은 알고리즘 함수 sort()를 사용한다.

===== 풀이 =====

  auto sortArr =(double arr[], int size){

sort(arr[0],arr[size-1]);};

1. pair & tuple 템플릿 클래스에 대하여 설명하시오

===== 풀이 =====

pair : 두 값을 그룹으로 묶는 클래스 , tuple : 여러 개를 하나로 묶어서 저장

1. 다음과 같이 선언된 map 객체 scores에 대하여 질문에 답하시오

map<string, int> scores;

1. scores 객체에 임의 원소를 추가하는 방법을 세가지 이상 제시하시오

===== 풀이 =====

scores.insert({“a”,0});

m[“b”] = 1;

m.try\_emplace(“c”,2);

1. scores 객체 저장된 모든 원소를 출력하는 방법을 두가지 이상 제시하시오

===== 풀이 =====

for(const auto &[key,value] : scores){ cout <<key<<value;}

for(const auto &v : scores){ cout<<v.frist<<v.second;

1. 람다식 구성에 대하여 설명하시오

===== 풀이 =====

캡쳐 리스트 :람다식에서 사용하고자 하는 함수 바깥의 변수 목록

매개변수 리스트 : 보통 함수의 매개변수 리스트와 동일

리턴 타입

함수 바디 : 람다식의 함수 코드

1. C++ STL 구성에 대하여 설명하시오

===== 풀이 =====

컨테이너 - 템플릿 클래스

iterator - 컨테이너 원소에 대한 포인터

알고리즘 - 템플릿 함수

* **적용 응용 프로그래밍 – 템플릿과 vector**

1. 배열의 순서를 역순으로 할 수 있는 할 수 있는 템플릿 클래스를 구현하고 제시된 결과처럼 실행될 수 있도록 프로그램을 완성 하시오. 단, 제시된 배열을 생성자 매개변수로 전달한다

template<typename T>

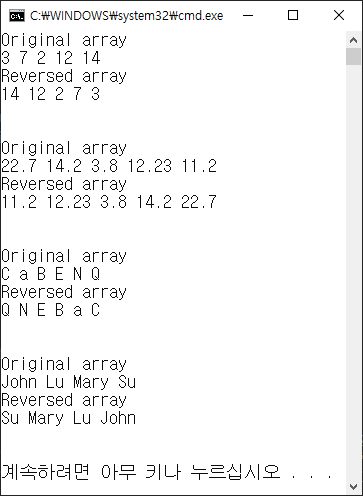
class Dim{

  T \*ptr; //T 타입의 배열을 가리키는 포인터

  int size; //배열 크기

  public:

  Dim(T\* arr, int size); //매개변수로 받은 배열과 배열크기를 멤버 변수로 대입

  void reverse(); //배열에 저장된 데이터 순서를 역순으로 저장

  void swap(T &first, T &second); //배열 원소 교환

  void print(); //배열 출력

};

int arr1[] = { 3, 7, 2, 12, 14 };

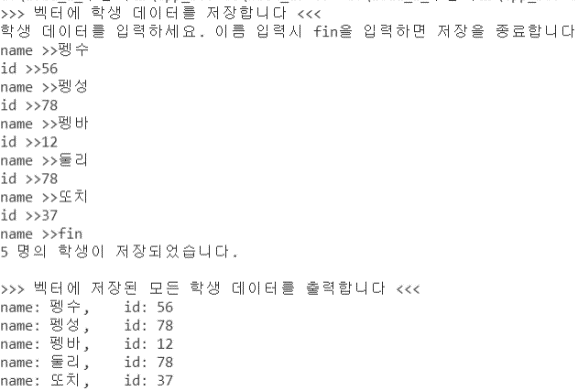
double arr2[] = { 22.7, 14.2, 3.8, 12.23, 11.2 };

char arr3[] = { 'C', 'a', 'B', 'E', 'N' , 'Q' };

string arr4[] = { "John" , "Lu", "Mary", "Su" };

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  using namespace std;  template <typename T>  class Dim  {  T \*ptr; // T 타입의 배열을 가리키는 포인터  int size; //배열 크기  public :  Dim(T \*arr, int size){ptr = arr, this->size = size;} //매개변수로 받은 배열과 배열크기를 멤버 변수로 대입  void reverse(); //배열에 저장된 데이터 순서를 역순으로 저장  void swap(T &first, T &second); //배열 원소 교환  void print(); //배열 출력  };  template <typename T>  void Dim<T>::reverse(){  int si = size/2;  for(int i =0; i<=si/2; i++){  swap(ptr[i],ptr[(size-1)-i]);  }  }  template <typename T>  void Dim<T>::swap(T &f, T&s){  T tmp;  tmp = f;  f = s;  s = tmp;  }  template <typename T>  void Dim<T>::print(){  cout <<"Original array"<<endl;  for(int i= 0; i <size; i++){  cout <<ptr[i]<<" ";  }  cout <<endl<<"Reversed array"<<endl;  reverse();  for (int i = 0; i < size; i++)  {  cout << ptr[i]<<" ";  }  cout <<endl<<endl;  }  int main(){  int arr1[] = {3, 7, 2, 12, 14};  double arr2[] = {22.7, 14.2, 3.8, 12.23, 11.2};  char arr3[] = {'C', 'a', 'B', 'E', 'N', 'Q'};  string arr4[] = {"John", "Lu", "Mary", "Su"};  Dim<int> a1(arr1,5);  Dim<double> a2(arr2, 5);  Dim<char> a3(arr3, 6);  Dim<string> a4(arr4, 4);  a1.print();  a2.print();  a3.print();  a4.print();  } |
| [실행 결과]  이미지 |

1. 벡터를 사용하여 학생 정보를 저장하고 출력하는 프로그램을 완성 하시오

class Student{

  string name;

  int id;

public:

  Student() = default;

  Student(string name, int id) : name(name), id(id){};

  ~Student() = default;

  void show()

  {

    cout << "name: " << name << ",    id: " << id << endl;

  }

};

class Manager{

  vector<Student> vec;

public:

  void run();

  void print();

  void save();

};

int main(){

  Manager man;

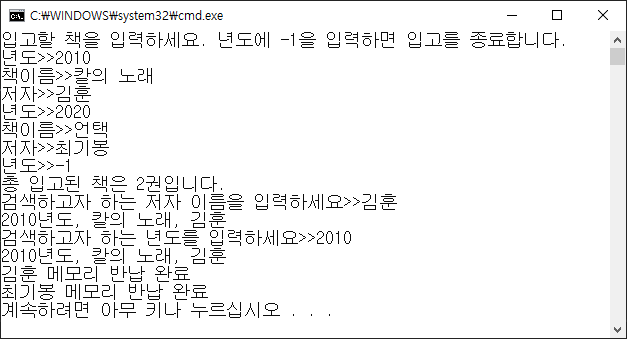
  man.run();

}

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  #include <vector>  using namespace std;  class Student  {  string name;  int id;  public:  Student() = default;  Student(string name, int id) : name(name), id(id){};  ~Student() = default;  void show()  {  cout << "name: " << name << ", id: " << id << endl;  }  };  class Manager  {  vector<Student> vec;  public:  void run();  void print();  void save();  };  void Manager::run()  {  cout<<">>> 벡터에 학생 데이터를 저장합니다. <<<"<<endl;  cout <<"학생 데이터를 입력하세요. 이름 입력시 fin을 입력하면 저장을 종료합니다."<<endl;  save();  print();  }  void Manager::save(){  string a;  int b;  while (true)  {  cout << "name >>";  cin >> a;  if (a == "fin")  break;  cout << "id :";  cin >> b;  vec.push\_back(Student(a, b));  }  cout << vec.size() << " 명의 학생이 저장되었습니다. " << endl;  }  void Manager::print()  {  cout << ">>> 백터에 저장된 모든 학생 데이터를 출력합니다 <<< "<<endl;  for(auto &a :vec){  a.show();  }  }  int main()  {  Manager man;  man.run();  } |
| [실행 결과]  이미지 |

1. 벡터를 사용하여 책 정보를 저장하고 저자와 년도로 검색하는 프로그램을 완성하시오

class Book {

 string title; // 책 이름

string author; // 저자

int year; //출판년도

public:

Book() {};

Book(string title, string author, int year);

string getAuthor();

int getYear();

void show(); //도서 정보 출력

}

};

class BookManager {

vector<Book\*> v; //Book 객체를 저장하기위한 vetor 객체 생성

void searchByAuthor(); //저자로 검색

void searchByYear(); //연도로 검색

void bookIn(); //벡터에 Book 정보 저장

void finish();

public:

void run();

};

void BookManager::run() {

bookIn(); // 입고

searchByAuthor(); // 저자로 검색

searchByYear(); // 년도로 검색

finish();

}

int main() {

BookManager man;

man.run();

}

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  #include <vector>  using namespace std;  class Book  {  string title; // 책 이름  string author; // 저자  int year; //출판년도  public:  Book(){};  Book(string title, string author, int year):title(title), author(author),year(year){};  string getAuthor(){return author;}  int getYear(){return year;}  void show(); //도서 정보 출력  };  void Book::show()  {  cout <<year<<"년도 , "<<title<<", "<<author<<endl;  }  class BookManager  {  vector<Book \*> v; // Book 객체를 저장하기위한 vetor 객체 생성  void searchByAuthor(); //저자로 검색  void searchByYear(); //연도로 검색  void bookIn(); //벡터에 Book 정보 저장  void finish();  public:  void run();  };  void BookManager::searchByAuthor(){  string a;  cout <<"검색하고자 하는 저자 이름을 입력하세요>>";  cin >> a;  for(auto &vec:v){  if(a==vec->getAuthor()){  vec->show();  break;  }  else{  cout <<"검색 결과가 없습니다."<<endl;  }  }  }  void BookManager::searchByYear()  {  int a;  cout << "검색하고자 하는 년도를 입력하세요>>";  cin >> a;  for (auto &vec : v)  {  if (a == vec->getYear())  {  vec->show();  break;  }  else  {  cout << "검색 결과가 없습니다." << endl;  }  }  }  void BookManager::bookIn()  {  int y;  string t, a;  cout << "입고할 책을 입력하세요. 년도에 -1을 입력하면 입고를 종료합니다."<<endl;  while(true){  cout <<"년도 >>";  cin >> y;  if(y == -1){  break;  }  cout << "책이름>>";  cin >> t;  cout <<"저자 >>";  cin >> a;  v.push\_back(new Book(t, a, y));  }  cout << "총 입고된 책은 "<<v.size()<<"권 입니다."<<endl;  }  void BookManager::finish()  {  for(auto &vv : v){  cout << vv->getAuthor()<< " 메모리 반납 완료"<<endl;  delete vv;  }  }  void BookManager::run()  {  bookIn(); // 입고  searchByAuthor(); // 저자로 검색  searchByYear(); // 년도로 검색  finish();  }  int main()  {  BookManager man;  man.run();  } |
| [실행 결과]  이미지 |

1. 배열에서 원소를 검색하는 search()함수를 템플릿으로 작성하시오. search()의 첫번째 매개변수는 검색하고자 하는 원소값이고 두번째 매개변수는 배열이며, 세번째 매개변수는 배열의 개수이다. 검색에 성공하면 true, 실패하면 false를 반환한다. 호출 사례는 다음과 같다. double, char 타입의 배열도 적용하여 결과를 제시하시오

int x[] = { 1, 10, 100, 5, 4 };

if (search(100, x, 5))

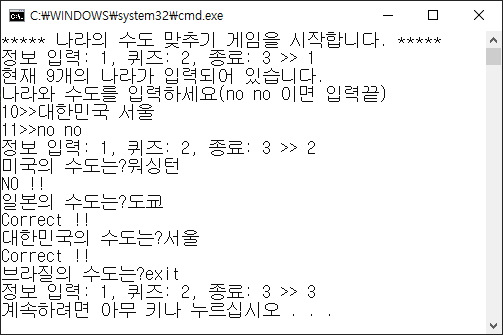
cout << "100이 배열 x에 포함되어 있다";

else

cout << "100이 배열 x에 포함되어 있지 않다";

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  using namespace std;  template <typename T>  bool search(T a,T b[], int c){  for(int i= 0; i <c; i++){  if(a == b[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";  else  cout << "100이 배열 x에 포함되어 있지 않다";  cout << endl;  double x2[] = {1.2, 10.3, 100.3, 5.4, 4.5};  if (search(100.6, x2, 5))  cout << "100이 배열 x에 포함되어 있다";  else  cout << "100이 배열 x에 포함되어 있지 않다";  cout <<endl;  char x3[] = {'1', '3', '9', '5', '4'};  if (search('5', x3, 5))  cout << "100이 배열 x에 포함되어 있다";  else  cout << "100이 배열 x에 포함되어 있지 않다";  } |
| [실행 결과]  이미지 |

1. 나라의 수도 맞추기 게임을 vector를 사용하여 작성하시오. 나라 이름과 수도 문자열로 구성된 Nation클래스를 만들고 vector<Nation> v;로 생성한 벡터를 이용하여 나라 이름과 수도 이름을 삽입할 수도 있고 랜덤하게 퀴즈를 볼 수도 있다. 프로그램 내에서 벡터에 Nation객체를 여러 개 미리 삽입하여 처리한다.



힌트)

Nation n[] = { Nation("미국", "와싱턴"), Nation("영국", "런던"), Nation("프랑스", "파리"),

Nation("중국", "베이찡"), Nation("일본", "도쿄"), Nation("러시아", "모스크바"),

Nation("브라질", "브라질리아"), Nation("독일", "베를린"), Nation("멕시코", "멕시코시티") };

for (int i = 0; i < 9; i++)

v.push\_back(n[i]);

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  #include <vector>  #include <ctime>  #include <cstdlib>  using namespace std;  class Nation{  string s1;  string s2;  public:  Nation(string s1, string s2):s1(s1),s2(s2){};  string getS1(){return s1;}  string getS2(){return s2;}  };  int main()  {  vector<Nation> v;  int count = 0;  bool flag = true;  string ans;  string j, k;  srand(time(nullptr));  Nation n[] = {Nation("미국", "와싱턴"), Nation("영국", "런던"), Nation("프랑스", "파리"),  Nation("중국", "베이찡"), Nation("일본", "도쿄"), Nation("러시아", "모스크바"),  Nation("브라질", "브라질리아"), Nation("독일", "베를린"), Nation("멕시코", "멕시코시티")};  for (int i = 0; i < 9; i++)  v.push\_back(n[i]);    cout << "\*\*\*\*\* 나라의 수도 맞추기 게임을 시작합니다. \*\*\*\*\*" << endl;  while (flag)  {  cout << "정보 입력 : 1, 퀴즈: 2, 종료: 3 >>";  cin >> count;  switch (count)  {  case 1:  cout << "현재 " << v.size()<<"개의 나라가 입력되어 있습니다."<<endl;  cout<<"나라와 수도를 입력하세요 (no no이면 입력끝)"<<endl;  while(true){  cout<< v.size()+1<<">>";  cin>>j>>k;  if(j=="no"&&k=="no")  break;  n[v.size() + 1] = Nation(j, k);  v.push\_back(n[v.size() + 1]);  }  break;  case 2:  while(true){  int b =rand()%v.size();  cout <<v[b].getS1()<<"의 수도는 ?";  cin>> ans;  if(ans == v[b].getS2()){  cout << "Correct !!"<<endl;  }  else if(ans == "exit"){  break;  }  else{  cout << "NO !!"<<endl;  }  }  break;  case 3:  flag = false;  break;  default:  cout <<"다시 입력해주세요 "<<endl;  break;  }  }  } |
| [실행 결과]  이미지 |