|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cpp\_10\_함수중복과 static멤버 | 학번 : | 20175260 | 이름 : | 최영빈 |

* **개념 확인 – 함수 중복과 static멤버**

1. 다음 질문에 O, X로 답하시오
   1. 함수 중복은 매개변수 타입이 다르거나 개수가 달라야 한다( o )
   2. 함수 중복은 리턴 타입만 달라도 가능하다 ( x )
   3. 소멸자는 매개변수를 가지지 않으므로 중복할 수 없다 ( o )
   4. 디폴트 매개변수 선언은 위치와 상관없다 ( x )
   5. 참조 매개변수로 인하여 모호성이 발생할 수 있기때문에 함수 중복 시 주의해야 한다( o )
   6. static 멤버 함수는 non-static 멤버는 사용할 수 없다( o )
   7. statc 멤버 함수에서 this는 선택적으로 사용할 수 있다 ( x )
   8. static 멤버는 클래스이름으로만 접근할 수 있다 ( x )
   9. static 멤버 선언 시 inline으로 선언하면 전역 공간에 선언할 필요가 없다( o )
   10. static 멤버는 객체 포인터로 접근할 수 있다 ( o )
2. 다음과 같은 디폴트 매개변수를 갖는 함수를 호출할 때 컴파일 오류가 발생하는 문장을 모두 고르시오

void msg(char id, int pass, string text="Hello“ );

1. msg('a', 20);
2. msg(); —————————오류
3. msg('b', 30, "default");
4. msg("C++"); —————————오류
5. msg('b', "default");—————————오류

1. 다음과 같이 중복된 함수를 실행한 결과를 출력 할 수 있는 문장을 모두 제시하시오
2. int sum(int a, int b, int c=30) {

return a + b + c;

}

==== 풀이 ====

sum(1,2,3); , sum(1,2);

1. double sum(double a=3.4, double b=6.5) {

return a + b;

}

==== 풀이 ====

sum(); sum(1.1); sum(1,2);

1. int sum(int a, int b=20) {

return a + b;

}

==== 풀이 ====

sum(1); sum(1,2);

1. Ex10 클래스의 static 멤버 함수를 호출하는 방법을 모두 제시하시오.

class Ex10 {

public:

static int cnt;

static void print();

};

==== 풀이 ====

객체 이름, 객체 포인터, 클래스명::

1. 위 5번 문제의 static 멤버 변수에 대한 외부 선언 문장을 제시하시오

==== 풀이 ====

Ex10::cnt

1. 위 5번 문제의 static 멤버 변수에 대한 외부 선언이 필요 없도록 선언하는 문장을 제시하시오

==== 풀이 ====

inline static int cnt;

1. 다음 두 개의 중복 함수를 디폴트 매개변수를 갖는 하나의 함수로 작성하시오

void fillLine(int n) {

for(int i=0; i<n; i++) cout << '※';

cout << endl;

}

void fillLine(int n, char c) {

for(int i=0; i<n; i++) cout << c;

cout << endl;

}

==== 풀이 ====

void fillLine(int n, char b = ‘※’)

1. 다음과 같이 선언된 함수에서 오류가 발생하는 원인을 제시하고 수정하시오

int calc(int a, int b=5, int c, int d=0)

==== 풀이 ====

디폴트 매개변수의 순서가 잘못되었다.

int calc(int a, int b, int c =5, int d =0)

1. 디폴트 매개변수에 대하여 설명하시오

==== 풀이 ====

매개 변수에 값이 넘어오지 않는 경우, 디폴트 값을 받도록 선언된 매개 변수

1. 함수 중복 모호성을 예제와 함께 설명하시오

==== 풀이 ====

float s(float a)

return a\*a;

double s(double a)

return a\*a;

int main()

cout << s(3); <——- int 타입 3을 double로 변환할지 float로 변환할 지 모호함

1. static 멤버와 non-static 멤버를 비교하여 설명하시오

==== 풀이 ====

static 멤버는 클래스 당 하나만 생성하며 클래스 멤버라고 불린다.

non-static 멤버는 객체마다 객체 내에 생성 , 인스턴스 멤버라고 불린다.

* **개념 적용 응용 프로그래밍 – 복사 생성자와 이동 생성자**

1. 책의 이름과 가격을 저장하는 Book클래스에 대하여 질문에 답하시오

class Book {

  char \*title; // 제목 문자열

  int price; // 가격

public:

  Book(const char\* title, int price);

  ~Book();

  void set(const char\* title, int price);

  void show() { cout << title << ' ' << price << "원" << endl; }

};

1. Book 클래스의 생성자, 소멸자, set()함수를 구현하시오. set() 함수는 멤버 변수 title에 할당된 메모리가 있으면 반환 후 새로운 메모리를 할당 받아 이곳에 매개변수로 전달 받은 책이름을 저장한다
2. 컴파일러가 삽입하는 디폴트 생성자 코드는 무엇인가? 복사생성자
3. 아래 main() 함수 실행 시 오류가 발생한다. 오류가 발생하지 않도록 복사 생성자를 추가하여 구현 하시오.

int main() {

  Book cpp("명품C++", 10000);

  Book java = cpp;

  java.set("명품자바", 12000);

  cpp.show();

  java.show();

}

1. main()에 다음과 같은 문장을 추가하였을 때 프로그램이 실행될 수 있도록 이동생성자를 추가 하시오

vector<Book> b;

b.push\_back(Book("명품파이썬", 300));

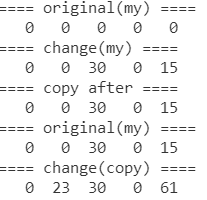
b.at(0).show();

Book book(Book{"명품스크립트", 34000});

book.show();

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  #include <vector>  #include <string.h>  using namespace std;  class Book  {  char \*title; // 제목 문자열  int price; // 가격  int size;  public:  Book(const char \*title, int price);  Book(const Book &b);  Book(Book && b) noexcept;  ~Book();  void set(const char \*title, int price);  void show() { cout << title << "" << price << "원" << endl; }  };  Book::Book(const char \*ti, int pr){  price = pr;  size = strlen(ti);  this->title = new char[size + 1];  strcpy(this->title, ti);  }  Book::Book(Book &&b) noexcept{  price = b.price;  title = b.title;  b.title = nullptr;  b.price = 0;  }  Book::~Book(){  delete []title;  }  Book::Book(const Book &b){  this->price = b.price;  size = strlen(b.title);  this->title = new char[size+1];  strcpy(this->title,b.title);  }  void Book::set(const char \*title, int price){  if(this->title){  delete[] this->title;  }  size = strlen(title);  this->price = price;  this->title = new char[size +1];  strcpy(this->title,title);    }  int main()  {  Book cpp("명품C++", 10000);  Book java = cpp;  java.set("명품자바", 12000);  cpp.show();  java.show();  vector<Book> b;  b.push\_back(Book("명품파이썬", 300));  [b.at](http://b.at)(0).show();  Book book(Book{"명품스크립트", 34000});  book.show();  } |
| [실행결과]  이미지 |

1. 제시된 결과처럼 실행될 수 있도록 main()함수를 참고하여 MyClass 클래스와 전역 함수 print()를 구현 하시오.



class MyClass{

  int size;

  int \*element;

  public:

  MyClass(int size);

  ~MyClass();

};

int main(){

  MyClass my{5};

  my.write("original(my)");

  my.change(2, 30);

  my.change(4, 15);

  my.write("change(my)");

  MyClass copy(my);

  copy.write("copy after");

  my.write("original(my)");

  copy.change(1, 23);

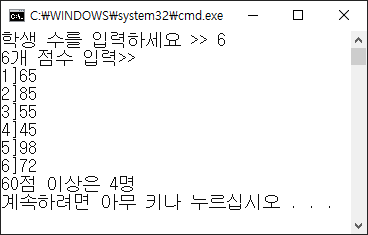
  copy.change(4, 61);

  print(move(copy), "change(copy)");

}

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  #include <iomanip>  using namespace std;  class MyClass{  int size;  int \*element;  public:  MyClass(int size);  ~MyClass();  MyClass(MyClass& my);  void write(string&& a);  void change(int a,int b);  };  MyClass::MyClass(int size) : size(size),element(new int[size]){  for(auto i = 0; i < size; i++){  element[i] = 0;  }  }  MyClass::MyClass(MyClass& my){  size = my.size;  element = new int[size];  for(auto i = 0; i <size; i++){  element[i] = my.element[i];  }  }  MyClass::~MyClass(){  if(!element)  delete[] element;  }  void MyClass::write(string&& a){  cout <<"==== "<< a<<" ===="<<endl;  for(int i = 0; i< size; i++){  cout << setw(4) << element[i];  }  cout <<endl;  }  void MyClass::change(int a, int b){  element[a] = b;  }  void print(MyClass &&m, string b);  void print(MyClass &&my, string b){  my.write(move(b));  }  int main()  {  MyClass my{5};  my.write("original(my)");  my.change(2, 30);  my.change(4, 15);  my.write("change(my)");  MyClass copy(my);  copy.write("copy after");  my.write("original(my)");  copy.change(1, 23);  copy.change(4, 61);  print(move(copy), "change(copy)");  } |
| [실행결과]  이미지 |

1. 한 분반의 인원수를 입력 받아 성적이 60이상인 학생의 수를 계산하는 프로그램을 제시된 main()과 결과를 참고하여 완성 하시오.



class Dept {

int size; // scores 배열의 크기

int\* scores; // 동적 할당 받을 정수 배열의 주소

public:

Dept(int size); //매개변수로 받은 배열 크기만큼 scores 배열 생성

Dept(Dept& dept); //복사 생성자

~Dept(); // 소멸자

int getSize(); //배열 크기 반환

void read(); // size 만큼 키보드에서 정수를 읽어 scores 배열에 저장

bool isOver60(int index); // index의 학생의 성적이 60보다 크면 true 리턴

};

int main() {

Dept \*com;

int cnt;

cout << "학생 수를 입력하세요 >> ";

cin >> cnt;

com = new Dept(cnt); //입력한 학생 수만큼 scores 배열 생성

com->read(); //학생들의 성적을 키보드로부터 읽어 scores 배열에 저장

int n = **countPass(\*com);** // com 학과에 60점 이상으로 통과한 학생의 수를 리턴

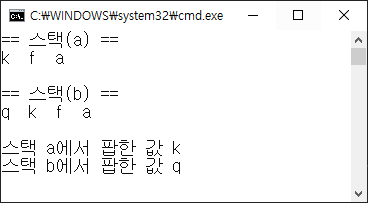
cout << "60점 이상은 " << n << "명" << endl;

delete com;

}

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  using namespace std;  class Dept  {  int size; // scores 배열의 크기  int \*scores; // 동적 할당 받을 정수 배열의 주소  public:  Dept(int size); //매개변수로 받은 배열 크기만큼 scores 배열 생성  Dept(Dept &dept); //복사 생성자  ~Dept(); // 소멸자  int getSize(); //배열 크기 반환  void read(); // size 만큼 키보드에서 정수를 읽어 scores 배열에 저장  bool isOver60(int index); // index의 학생의 성적이 60보다 크면 true 리턴  };  Dept::Dept(int size){  this->size = size;  scores = new int[size];  }  Dept::Dept(Dept &dept){  this->size = dept.size;  for(auto i =0; i< size; i ++){  scores[i] = dept.scores[i];  }  }  Dept::~Dept(){  delete[] scores;  }  int Dept::getSize(){  return size;  }  void Dept::read(){  cout << size << "개 점수 입력 >> " << endl;  for(auto i =0; i < size; i++){  cout <<i+1<<"] ";  cin >> scores[i];  }  }  bool Dept::isOver60(int index){  if(scores[index] > 60)  return true;  return false;  }  int countPass(Dept &a);  int countPass(Dept &a){  int count = 0;  for(int i = 0; i < a.getSize(); i++ ){  bool flag = a.isOver60(i);  if(flag)  count++;  }  return count;  }  int main()  {  Dept \*com;  int cnt;  cout << "학생 수를 입력하세요 >> ";  cin >> cnt;  com = new Dept(cnt); //입력한 학생 수만큼 scores 배열 생성  com->read(); //학생들의 성적을 키보드로부터 읽어 scores 배열에 저장  int n = countPass(\*com); // com 학과에 60점 이상으로 통과한 학생의 수를 리턴  cout << "60점 이상은 " << n << "명" << endl;  delete com;  } |
| [실행결과]  이미지 |

1. 문자를 저장할 수 있는 스택 클래스 MyInStack을 구현하고 제시된 결과처럼 실행될 수 있도록 프로그램을 완성 하시오. 단, 스택에 저장할 수 있는 최대 문자수는 생성자 매개변수로 주어지고 size멤버에 저장한다.



class MyIntStack {

char \*p; // 스택 메모리로 사용할 포인터

int size; // 스택의 최대 크기

int tos; // 스택의 탑을 가리키는 인덱스

public:

MyIntStack();

MyIntStack(int size);

MyIntStack(MyIntStack& s);

~MyIntStack();

bool push(char n); // 정수 n을 스택에 푸시,스택이 꽉 차 있으면 false를, 아니면 true 리턴

bool pop(char &n); // 스택의 탑에 있는 값을 n에 팝, 만일 스택이 비어 있으면 false를, 아니면 true 리턴

void show(); //스택에 저장된 모든 문자 출력

};

int main() {

MyIntStack a(20);

a.push('a');

a.push('f');

a.push('k');

cout << "== 스택(a) ==" << endl;

a.show();

MyIntStack b = a; // 복사 생성

cout << endl<<"== 스택(b) ==" << endl;

b.push('q');

b.show();

char n;

a.pop(n); // 스택 a 팝

cout << endl<<"스택 a에서 팝한 값 " << n << endl;

b.pop(n); // 스택 b 팝

cout << "스택 b에서 팝한 값 " << n << endl<<endl;

}

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string.h>  #include <string>  #include <iomanip>  using namespace std;  class MyIntStack  {  char \*p; // 스택 메모리로 사용할 포인터  int size; // 스택의 최대 크기  int tos; // 스택의 탑을 가리키는 인덱스  public:  MyIntStack(){size = 0, tos = -1;}  MyIntStack(int size);  MyIntStack(MyIntStack &s);  MyIntStack(MyIntStack &&s);  ~MyIntStack();  bool push(char n);  bool pop(char &n);  void show();  };  MyIntStack::MyIntStack(int size){  this->size = size;  tos = -1;  p = new char[size];  }  MyIntStack::MyIntStack(MyIntStack &s){  this->size = s.size;  tos = s.tos;  p = new char[size];  strcpy(p,s.p);  }  MyIntStack::MyIntStack(MyIntStack &&s) {  size = s.size;  tos = s.tos;  p = s.p;  s.size = 0;  s.tos = 0;  s.p = nullptr;  }  MyIntStack::~MyIntStack(){  delete[] p;  }  bool MyIntStack::push(char n){  if(size == tos)  return false;  tos++;  p[tos] = n;  return true;  }  bool MyIntStack::pop(char &n){  if(tos == 0)  return false;  n = p[tos];  return true;  }  void MyIntStack::show(){  for(int i =size; i >= 0; i--){  if(p[i] != '\0')  cout <<setw(4) <<p[i];  }  cout <<endl;  }  int main()  {  MyIntStack a(20);  a.push('a');  a.push('f');  a.push('k');  cout << "== 스택(a) ==" << endl;  a.show();  MyIntStack b = a; // 복사 생성  cout << endl  << "== 스택(b) ==" << endl;  b.push('q');  b.show();  char n;  a.pop(n); // 스택 a 팝  cout << endl  << "스택 a에서 팝한 값 " << n << endl;  b.pop(n); // 스택 b 팝  cout << "스택 b에서 팝한 값 " << n << endl  << endl;  } |
| [실행결과]  이미지 |