|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2021\_1\_C++\_03\_개념확인 | 학번 : | 20185309 | 이름 : | 황명원 |

* 기본 개념 확인

1. 빈 괄호를 채워 넣으시오.
2. ( 클래스 )란 객체를 만들기 위해 정의된 설계도이다
3. C++에서 클래스는 클래스 선언부와 클래스 ( 구현부 )로 구성된다
4. 클래스 선언시 멤버에 대한 접근 권한이 생략되면 디폴트는 ( private ) 이다.
5. ( 생성자 )는 객체가 생성되는 시점에서 자동으로 호출되는 멤버함수이다
6. 생성자가 선언되어 있지 않으면 ( 컴파일러 )에 의해 자동으로 생성된다.
7. 클래스 멤버의 디폴트 접근권한은 (private )이며 구조체 디폴트 접근 권한은 ( public )이다
8. 한정자 ( const )를 사용하여 멤버를 상수화 할 수 있다
9. 인라인 함수는 함수 호출에 따른 (시간 )를 줄이기 위한 방법이다.
10. ( mutable ) 키워드는 상수화 된 대상에 대한 쓰기 작업을 허용하기 위한 목적으로 사용한다
11. 함수를 인라인으로 하려면 키워드 ( inline )을 사용한다.
12. 다음 질문에 O, X로 답하시오
13. 소멸자는 여러 번 구현할 수 있다 (x )
14. 생성자는 반환값이 없다 ( o )
15. 프로그램이 종료하면 전역 객체가 생성된 순서로 객체가 소멸된다 ( x )
16. private 접근 권한은 생성자에 사용할 수 없다 ( x )
17. 인라인 함수는 inline 키워드로 선언된 함수이다 (o )
18. 생성자는 객체 생성 시 한번만 호출할 수 있다 ( x )
19. 소멸자는 매개변수를 가질 수 없다( o )
20. 생성자와 소멸자의 실행 순서는 동일하다 ( x )
21. 인라인 선언은 크기가 큰 함수의 경우 효과적이다 (x )
22. 인라인 함수를 사용하면 프로그램 크기가 작아져서 효과적이다 ( x )

1. 질문에 답하시오
2. 다음과 같은 생성자와 소멸자가 있다고 할 때 어떤 것이 디폴트 생성자인지, 소멸자인지, 복사 생성자인지 구분하시오

Account (); 생성자

**~**Account (); 소멸자

Account (const Account & acc); 복사생성자

|  |
| --- |
| **int** Rectangle **(int** length, **int** height); |

1. 다음과 같이 Rectangle 클래스의 생성자를 선언하면 오류가 발생한다. 이유를 설명하시오

생성자는 리턴타입이 없기때문에 앞에 int를 붙이지않는다(void도 마찬가지)

고칠려면 Rectangle(int length,int height);

1. 다음과 같이 Rectangle 클래스의 소멸자를 선언하면 오류가 발생한다. 이유를 설명하시오.

|  |
| --- |
| **int** ~Rectangle(int var); |

소멸자는 리턴타입이 없고 어떤값도 리턴하면 안되고 매개변수가 없으므로 ~Rectangle();

1. 다음과 같은 생성자를 할당하는 코드 대신 초기화 리스트를 사용하는 코드로 변경하세요.

Rectangle :: Rectangle (int len, int wid){

length = len;

height = wid;

}

->Rectangle::Rectangle(int len, int wid) : length(len),height(wid)

1. 다음과 같은 클래스 정의가 있을 때 객체 생성 코드 중 오류가 발생하는 것을 제시하시오

class Sample{

private :

int x;

public:

int getX() const;

};

Sample s1(4);

Sample s2(); //오류

Sample s3{};

1. 다음 클래스 선언에서 오류가 발생하는 코드를 수정하시오

class Object{

double x;

public; -> public: //수정

double const getX(); -> double getX() const; //수정

};

1. x가 클래스의 데이터 멤버라고 할 때, 다음 클래스의 멤버 함수 정의에서 발생하는 오류를 수정 하시오.

Member::int getName(){ -> int Member::getName() 으로 수정

return name;

}

1. 다음과 같은 클래스 선언에 대하여 생성자와 접근자 함수를 구현 하시오

class Hallym{

string dept;

public:

Hallym(string d); //매개변수값을 멤버 변수로 초기화

string getDept(); //멤버 변수 값 반환

};

Hallym::Hallym(string d){

dept=d;

}

string Hallym::getDept(){

return dept;

}

1. 다음과 같은 멤버를 갖는 Person 클래스를 인터페이스, 구현, 애플리케이션 코드로 각각 분리하여 프로그램을 작성하시오.

|  |
| --- |
| * 데이터 멤버 - name과 age * 접근자 멤버 함수 - getName과 getAge * 설정자 멤버 함수 - setName과 setAge * 매개변수가 있는 생성자와 소멸자   소스코드  -인터페이스  // // Created by 황명원 on 2021/03/20. //  #ifndef BASIC03\_19\_PERSON\_H #define BASIC03\_19\_PERSON\_H  #include <string> #include <iostream> using namespace std; using std::string;  class Person{   string name;  int age; public:   Person(string n,int a);  ~Person();  string getName();  int getAge();  void setName(string n);  void setAge(int a);  void print(); };   #endif //BASIC03\_19\_PERSON\_H  -구현   1. 접근자 멤버함수   #include "Person.h"  string Person::getName() {  return name; } int Person::getAge() {  return age; }   1. 설정자 멤버함수   #include "Person.h"  void Person::setName(string n){  name=n; } void Person::setAge(int a) {  age=a; }   1. 매개변수가있는 생성자와 소멸자   #include "Person.h"  Person::Person(string n, int a) {  name=n;  age=a; } Person::~Person() { }  -애플리케이션 코드  #include <iostream> using namespace std; #include "Person.h"  int main() {  Person per("명원",23);  cout<<"이름 "<<per.getName()<<" 나이:"<<per.getAge()<<endl;   per.setName("정원");  per.setAge(19);   cout<<"이름 "<<per.getName()<<" 나이:"<<per.getAge()<<endl;  }  -실행결과  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. 제시된 main()소스코드를 참조하여 계좌 관리를 위한 클래스 Account를 구현하시오. 단, 클래스 구현과 선언을 분리하여 작성하시오

enum MENU { DEPOSIT = 1, WITHDRAW, INQUIRY };

|  |
| --- |
| int main() {  Account a("C++", 2021, 50000);  int menu, money;  cout << "menu : 1. 입금, 2. 출금 3. 조회 >> ";  cin >> menu;  switch (menu) {  case MENU::DEPOSIT:  cout << "입금액 >> ";  cin >> money;  a.deposit(money);  cout << a.getOwner() << "의 입금 액은 " << money << endl;  cout << a.getOwner() << "의 잔액은 " << a.inquiry() << endl;  break;  case MENU::WITHDRAW :  cout << "출금액 >> ";  cin >> money;  cout << a.getOwner() << "의 출금 액은 " << a.withdraw(money) <<endl;  cout << a.getOwner() << "의 잔액은 " << a.inquiry() << endl;  break;  case MENU::INQI:  cout << a.getOwner() << "의 잔액은 " << a.inquiry() << endl;  }  return 0;  } |

소스코드

-인터페이스

#ifndef BASIC3\_20\_ACCOUNT\_H  
#define BASIC3\_20\_ACCOUNT\_H  
#include <string>  
#include <iostream>  
using namespace std;  
  
class Account{  
 int menu,money;  
 string name;  
 int id;  
 int balance;  
public:  
  
 Account(string n,int i,int mny);  
 ~Account();  
 void deposit(const int money);  
 int withdraw(int money);  
 int inquiry() const;  
 string getOwner() const;  
};  
#endif //BASIC3\_20\_ACCOUNT\_H

-구현

#include "Account.h"  
  
Account::Account(string n, int i, int mny) {  
 name=n;  
 id=i;  
 balance=mny;  
  
}  
Account::~Account() {  
 cout<<id<<": 객체 소멸"<<endl;  
}  
  
void Account::deposit(const int money) {  
 balance+=money;  
}  
int Account::withdraw(int money) {  
 if(money>balance){  
 money=balance;  
 balance;  
 }  
 else  
 balance-=money;  
 return money;  
}  
int Account::inquiry() const {  
 return balance;  
}  
string Account::getOwner() const {  
 return name;  
}

-애플리케이션 코드

#include <iostream>  
#include "Account.h"  
using namespace std;  
  
enum MENU{*DEPOSIT*=1,*WITHDRAW*,*INQUIRY*};  
  
int main() {  
 Account a("C++", 2021, 50000);  
 int menu, money;  
 cout << "menu : 1. 입금, 2. 출금 3. 조회 >> ";  
 cin >> menu;  
 switch (menu) {  
 case MENU::*DEPOSIT*:  
 cout << "입금액 >> ";  
 cin >> money;  
 a.deposit(money);  
 cout << a.getOwner() << "의 입금 액은 " << money << endl;  
 cout << a.getOwner() << "의 잔액은 " << a.inquiry() << endl;  
 break;  
 case MENU::*WITHDRAW* :  
 cout << "출금액 >> ";  
 cin >> money;  
 cout << a.getOwner() << "의 출금 액은 " << a.withdraw(money) <<endl;  
 cout << a.getOwner() << "의 잔액은 " << a.inquiry() << endl;  
 break;  
 case MENU::*INQUIRY*:  
 cout << a.getOwner() << "의 잔액은 " << a.inquiry() << endl;  
 }  
 return 0;  
}

-실행 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 장치, 측정기, 게이지이(가) 표시된 사진

자동 생성된 설명

텍스트, 장치, 측정기, 게이지이(가) 표시된 사진

자동 생성된 설명