

1.

헤더파일

여러가 발생하는 것을.

(1) 전처리는 해당 전처리가 쓰인 헤더파일이 한 번만 보드되도록 한다. 여러번 정의되는 것 즉 다중정의가 돼 방지할 수 있다.

(2) this 포인터는 객체 내부에서 자기 자신을 가리키는 포인터 역할을 한다.

this는 암묵적으로, 명시적으로 사용 가능하다. (언제까지 않으면 this를 암묵적으로 사용한 것이다.)

this를

(3) 생성자의 인자가 주지 않음 때에도 데이터 멤버를 일관성 있게 초기화 한다. 또 디폴트 인자가 지정된 생성자는 디폴트 생성자이다. 따라서 어떠한 인자도 없이 호출하는 것도 가능하다. (디폴트 생성자도 이용 가능)

(4) 정적은 어떤 클래스가 다른 클래스의 객체를 멤버로 갖는 것으로 소프트웨어의 재사용성의 형태를 띈다.

마리

(5) 재사용성이 뛰어나다. 클래스를 상속해놓으면 새로운 클래스를 작성, 또는 프로그램 작성시 기존 클래스의 재사용을 하면 시간과 노력을 줄 수 있다. 또한 재사용은 프로그래머가 더 신뢰있고 효율적인 시스템을 구축할 수 있게 한다.

또 캡슐화를 통해 정보보호가 가능하며 잘 정의된 인터페이스를 통한 통신이 가능하다.

2. (1) X

(2) O

(3) O

(4) O

(5) O.

(6) X.

(7) X.

(8) X

(9) X

(10) O

(11) X

(12) X

(13) O

(14) X

(15) O.

3. (1) const 변수를 초기화 때는 멤버 변수라야지 써야함.

noon() : h(12), m(0), s(0) { } 으로 수정.

(2) ① 매진 parameter에 디폴트값이 지시안돼 오류 발생

Time (int=0, int=0, int=0); 수정

③ class 정의 { } 뒤에 ; 붙이기.

② 포인터 객체는 객체 메모리에 접근시 → 이용해야함
ClockPer → printUniversal(); 수정.

```
class Time
{
:
};
```

(3).

① (*clockPer).second=0; 으로 수정.

<생략>

4.



100 줄까지

다시.

5.

<생략>

date 생략자
Em 생략자
date 생략자
date 생략자
date 생략자
Em 생략자.
Em 생략자
date 생략자
Em 생략자
date 생략자
date 생략자
date 생략자

