

OOP 학기 지필평가

1. Primitive Type 에 관한 내용 중, 옳은 것을 모두 고르세요. 없으면 '없음' 으로 답하세요

- 1) char 타입은 int 타입과 변환이 불가능하다
- 2) 문자 '1' 은 정수타입 1 과 같다
- 3) 타입의 표현범위를 넘어서는 연산을 해도 *오류가 발생하지 않는다
- 4) 정수타입 대신 enum 타입을 사용하면 프로그램의 성능이 개선된다
- 5) 수식에서 산술연산자는 우선순위와 관계없이 왼쪽부터 계산한다

2. 아래처럼 타입이 정의 되어 있습니다.

```
class A {  
    public int Number = 0;  
}
```

2 - 1. 아래 코드의 실행을 예상하여 출력 결과를 적으세요

```
A a1 = new A(){ Number = 1 };  
A a2 = new A(){ Number = 1 };  
  
if( a1 == a2 )  
    Console.WriteLine("a1 is a2");  
else  
    Console.WriteLine("a1 is not a2");
```

2 - 2. 아래 코드에서 *오류가 발생하는 지점이 있다면 설명하고, 없다면 '없음' 으로 답하세요

```
A a3 = null;  
  
if( a3.Number == 0 || a3 != null )  
{  
    Console.WriteLine($"Number of a3 is {a3.Number}");  
}
```

3. 아래 코드에서 *오류가 발생하는 지점이 있다면 설명하고, 없다면 '없음' 으로 답하세요.

```
for (int i = 0; ; i++)  
{  
    Console.WriteLine(i);  
}
```

4. 객체 지향 프로그래밍에 관한 설명 중, 틀린것을 모두 고르세요. 없으면 '없음' 으로 답하세요

- 1) 자동차, 의자, 사물 등 시각적으로 구분 할 수 있는 것만 객체로 정의가 가능하다
- 2) 실재하는 대상을 객체로 다룰 때는 관점이 달라도 의미는 항상 유지된다
- 3) 절차지향 프로그래밍에 비해 성능이 좋기 때문에 선호된다
- 4) 객체를 개별적으로 수정하긴 어렵지만 재사용성이 용이하다
- 5) 추상 클래스는 인스턴스화 되지 않는다

5. 아래처럼 타입이 정의되어 있습니다

<pre>class A { private int _number = 0; public int Number => _number; (2) public void Add(B b) { _number += b.Number; } public void Hellow() {} }</pre>	<pre>class B { private int _number = 0; public int Number => _number; (2) public void Add(A a) { _number += a.Number; } public void Byebye() {} }</pre>
--	--

5 - 1. 상속의 개념을 활용하여 개선하세요

5 - 2. 코드블럭(2) 가 포함되도록 인터페이스를 활용하여 개선하세요

6. 아래처럼 타입이 정의되어 있습니다

```
interface IHandshake{  
    public void Hi();  
    public void Bye();  
}
```

6 - 1. IHandshake 를 상속한 어떤 클래스 A 에 대해, 아래 코드에서 틀린 부분을 모두 고르세요. 없으면 '없음' 으로 답하세요. (틀린 부분이 있는경우, 그 이유도 서술하세요)

```
IHandshake shaker1 = new A();  
shaker1.Hi();  
A shaker2 = new A();  
shaker2.Bye();  
(shaker2 as IHandshake).Hi();  
A shaker3 = new IHandshake();  
shaker3.Hi();  
IHandshake ishaker2 = shaker2;  
ishaker2.Bye();
```

6 - 2. HowAreYou() 메소드를 가진 클래스 A 와, FineThankAndYou() 메소드를 가진 클래스 B 가 모두 IHandshake 를 상속하도록 정의하세요

6 - 3. 6-2 에서 정의한 클래스가 아래 코드블럭에서 동작하도록 빈 칸 **[-n-]** 을 채우세요

```
var handshakers = new List<[-1-]>();  
handshakers.Add([-2-]);  
handshakers.Add([-3-]);  
  
for(int i = 0; i < handshakers.Count ; i++)  
    handshakers[i].Hi();  
  
([-4-]).HowAreYou();  
  
var handshaker2 = handshake[1];  
if ( handshaker2 [-5-] )  
    ( handshaker2 [-6-] ).FineThankAndYou();
```

* 오류 : Exception 이 발생하는 경우만 해당. 무한루프 등의 버그는 여기서는 예외로 합니다.

객체지향 프로그래밍 기말고사 답안지

이름 :
학번 :

* 답안지의 양면을 모두 활용하여도 좋습니다.