Question 1

Q: 파이썬에서 클래스**(class)**란 무엇인가요**?**

- A) 함수
- B) 모듈
- C) 코드 블록
- D) 객체(object)를 생성하기 위한 청사진

Question 2

Q: 파이썬에서 `__metaclass__` 속성의 역할은 무엇인가요**?**

- A) 클래스 속성을 정의하기 위해
- B) 클래스의 슈퍼클래스를 지정하기 위해
- C) 클래스의 메타클래스를 나타내기 위해
- D) 인스턴스에 새로운 속성 생성을 제한하기 위해

Question 3

Q: 캡슐화**(encapsulation)**의 맥락에서**,** 파이썬의 접근 지정자**(access modifiers)**의 목적은 무엇인가요**?**

- A) 클래스 속성을 정의하기 위해
- B) 객체의 특정 구성 요소에 대한 접근을 제한하기 위해
- C) 읽기 전용 속성을 생성하기 위해
- D) 인스턴스 변수를 초기화하기 위해

Question 4

Q: OOP에서 동적 다형성**(dynamic polymorphism)**이란 무엇인가요**?**

- A) 런타임에 객체를 생성하는 능력
- B) 변수의 타입을 동적으로 변경하는 능력
- C) 런타임에 다른 타입들을 위한 단일 인터페이스를 제공하는 능력
- D) 클래스에 다른 이름의 메소드를 정의하는 능력

Question 5

Q: OOP에서 다중 상속**(multiple inheritance)**의 주된 목적은 무엇인가요**?**

- A) 복잡한 클래스 계층을 생성하기 위해
- B) 여러 클래스의 속성을 결합하기 위해
- C) 클래스가 하나 이상의 슈퍼클래스로부터 상속받을 수 있게 하기 위해
- D) 여러 클래스의 메소드를 오버라이드(override)하기 위해

Question 6

Q: 캡슐화**(encapsulation)**는 **OOP**에서 데이터 보호에 어떻게 기여하나요**?**

- A) 모든 클래스 속성을 공개적으로 노출함으로써
- B) private 속성에 대한 직접 접근을 허용함으로써
- C) 객체의 특정 구성 요소에 대한 접근을 제한함으로써
- D) 클래스 멤버에 대한 전역(global) 접근을 조장함으로써

Question 7

Q: 파이썬 클래스에서 `__init__` 메소드의 목적은 무엇인가요**?**

- A) 인스턴스 변수 초기화하기
- B) 인스턴스 변수 선언하기
- C) 클래스를 위한 메소드 정의하기
- D) 새로운 클래스 생성하기

Question 8

Q: 파이썬에서 캡슐화**(encapsulation)**를 어떻게 구현할 수 있나요**?**

- A) private 변수와 메소드를 사용함으로써
- B) 클래스에 같은 이름의 여러 메소드를 정의함으로써
- C) 클래스의 새 인스턴스를 생성함으로써
- D) `@property` 데코레이터를 사용함으로써

Question 9

Q: 파이썬에서 `__str__` 메소드의 목적은 무엇인가요**?**

- A) 객체를 문자열 표현으로 변환하기 위해
- B) 클래스에 추상 메소드를 정의하기 위해
- C) 객체가 더 이상 필요하지 않을 때 정리 활동을 수행하기 위해
- D) 인스턴스 변수를 초기화하기 위해

Question 10

Q: `abc` 모듈을 사용하여 파이썬에서 추상 클래스**(abstract class)**를 어떻게 생성할 수 있나요**?**

- A) `abstract` 키워드를 사용함으로써
- B) `@abstractmethod` 데코레이터로 메소드를 정의함으로써
- C) 클래스 정의에 `abstract` 속성을 사용함으로써
- D) 구체적인(concrete) 메소드가 없는 클래스를 생성함으로써

Question 11

Q: 파이썬에서 `@property` 데코레이터의 목적은 무엇인가요**?**

- A) 클래스 속성을 정의하기 위해
- B) 읽기 전용 속성을 생성하기 위해
- C) 쓰기 전용 속성을 생성하기 위해
- D) 클래스 메소드를 생성하기 위해

Question 12

Q: 파이썬에서 메소드 오버로딩**(method overloading)**을 어떻게 달성할 수 있나요**?**

- A) 클래스에 같은 이름의 여러 메소드를 정의함으로써
- B) 단일 메소드에서 기본 인자(default arguments)나 `*args`, `**kwargs`를 사용함으로써
- C) `super()` 함수를 사용함으로써
- D) 클래스에 같은 매개변수를 가진 여러 메소드를 정의함으로써

Question 13

Q: OOP에서 추상 클래스**(abstract class)**란 무엇인가요**?**

- A) 인스턴스화할 수 없는 클래스
- B) 인스턴스화할 수는 있지만 상속될 수 없는 클래스
- C) 추상 메소드를 정의하고 구체적인(concrete) 메소드를 가질 수도 있는 클래스
- D) 어떠한 메소드도 가질 수 없는 클래스

Question 14

Q: 파이썬에서 메타클래스**(metaclass)**란 무엇인가요**?**

- A) 여러 클래스로부터 상속받는 클래스
- B) 클래스를 생성하기 위한 클래스
- C) 인스턴스화할 수 없는 클래스
- D) 슈퍼클래스의 모든 메소드를 오버라이드하는 클래스

Question 15

Q: 다형성**(polymorphism)**은 **OOP**에서 코드 유연성에 어떻게 기여하나요**?**

- A) 여러 클래스를 하나로 결합함으로써
- B) 다른 타입들을 위한 단일 인터페이스를 제공함으로써
- C) 객체의 특정 구성 요소에 대한 접근을 제한함으로써
- D) 같은 이름의 여러 메소드를 정의함으로써

Question 16

Q: OOP에서 캡슐화**(encapsulation)**를 사용했을 때의 이점은 무엇인가요**?**

- A) 향상된 코드 구성
- B) 코드 재사용성
- C) 향상된 코드 가독성
- D) 향상된 시스템 성능

Question 17

Q: 기존 클래스를 기반으로 새 클래스를 생성하는 과정을 무엇이라고 하나요**?**

- A) 캡슐화 (Encapsulation)
- B) 추상화 (Abstraction)
- C) 상속 (Inheritance)
- D) 다형성 (Polymorphism)

Question 18

Q: OOP에서 추상화**(abstraction)**의 주된 목적은 무엇인가요**?**

- A) 여러 클래스를 하나로 결합하는 것

- B) 객체의 특정 구성 요소에 대한 접근을 제한하는 것
- C) 클래스에 같은 이름의 여러 메소드를 정의하는 것
- D) 필수적인 특성을 기반으로 클래스를 모델링하여 복잡한 시스템을 단순화하는 것

Question 19

Q: 파이썬에서 메소드 오버로딩**(method overloading)**을 달성하기 위해 사용되는 키워드는 무엇인가요**?**

- A) overload
- B) method
- C) override
- D) 파이썬은 메소드 오버로딩을 지원하지 않습니다

Question 20

Q: OOP에서 다형성**(polymorphism)**이란 무엇인가요**?**

- A) 여러 클래스를 하나로 결합하는 것
- B) 객체의 특정 구성 요소에 대한 접근을 제한하는 것
- C) 클래스에 같은 이름의 여러 메소드를 정의하는 것
- D) 다른 타입들을 위한 단일 인터페이스를 제공하는 것

Question 21

Q: OOP에서 상속**(inheritance)**은 클래스가 무엇을 할 수 있게 하나요**?**

- A) 여러 클래스를 하나로 결합
- B) 같은 이름의 여러 메소드를 정의
- C) 슈퍼클래스로부터 속성과 행동을 상속
- D) 객체의 특정 구성 요소에 대한 접근을 제한

Question 22

Q: 클래스 변수**(class variable)**와 인스턴스 변수**(instance variable)**의 차이점은 무엇인가요**?**

- A) 클래스 변수는 인스턴스에 한정되지만, 인스턴스 변수는 모든 인스턴스간에 공유됩니다.
- B) 클래스 변수는 클래스 외부에 정의되지만, 인스턴스 변수는 클래스 내부에 정의됩니다.

- C) 클래스 변수는 모든 인스턴스간에 공유되지만, 인스턴스 변수는 각 인스턴스에 한정됩니다.
- D) 클래스 변수는 메소드 오버로딩에 사용되지만, 인스턴스 변수는 메소드 오버라이딩에 사용됩니다.

Question 23

Q: OOP에서 캡슐화**(encapsulation)**란 무엇인가요**?**

- A) 클래스를 모듈로 변환하는 과정
- B) 여러 클래스를 단일 클래스로 결합하는 것
- C) 객체의 특정 구성 요소에 대한 접근을 제한하고 구현 세부 사항을 숨기는 것
- D) 클래스의 새 인스턴스를 생성하는 것

Question 24

Q: 클래스에서 인스턴스 변수**(instance variable)**란 무엇인가요**?**

- A) 클래스의 모든 인스턴스간에 공유되는 변수
- B) 클래스 외부에 정의된 변수
- C) 클래스의 인스턴스에 한정된 변수
- D) 메소드 오버로딩에 사용되는 변수

Question 25

Q: 객체 지향 프로그래밍**(OOP)**의 주된 목표는 무엇인가요**?**

- A) 코드 재사용성 및 모듈성
- B) 효율적인 메모리 활용
- C) 절차적 추상화
- D) 동적 타이핑

Question 26

Q: 파이썬 이터레이터**(iterator)**에서 `__next__()` 메소드는 무엇을 하나요**?**

- A) 이터레이터를 초기화합니다
- B) 다음 항목을 반환하거나 남은 항목이 없으면 StopIteration을 발생시킵니다
- C) 이터레이터 객체 자체를 반환합니다
- D) 이터러블(iterable)을 리스트로 변환합니다

Question 27

Q: 파이썬 이터레이터**(iterators)**와 이터러블**(iterables)**에 대한 다음 설명 중 옳은 것은 무엇인가요**?**

- A) 이터러블은 `__next__()`를 구현해야 합니다
- B) 이터레이터는 for 루프에서 사용될 수 없습니다
- C) 이터레이터는 `__iter__()`와 `__next__()`를 모두 구현합니다
- D) 이터러블은 자동으로 StopIteration을 발생시킵니다

정답

D C B C C
C A A A B
B B C B B
A C D D C
C C C C A
B C