

1. 자바의 연산자 중에 나머지 연산자는?

- ① *
- ② /
- ③ %
- ④ &

2. 다음 중 자바로 “Hello”를 출력하는 프로그램은?

- ① System.out.println(“Hello”);
- ② printf(“%s\\n”, “Hello”);
- ③ Console.WriteLine(“Hello”);
- ④ 위 모두 다 가능함

3. 자바 프로그램을 만드는 과정의 순서가 올바른 것을 고르시오.

- ① 프로그램 입력 - 컴파일 - 디버깅 - 실행
- ② 컴파일 - 프로그램 입력 - 디버깅 - 실행
- ③ 프로그램 입력 - 디버깅 - 컴파일 - 실행
- ④ 디버깅 - 컴파일 - 실행 - 프로그램 입력

4. 다음 표의 빈간에 7개의 기본형(char, byte, short, int, long, float, double)을 boolean처럼 알맞은 자리에 넣으시오.

	1바이트	2바이트	4바이트	8바이트
논리형	boolean			
문자형				
정수형				
실수형				

5. 5 & 2의 결과 값은?

- ① 0
- ② 2
- ③ 3

6. 오버라이딩을 막는 방법의 선언문은 무엇인가?

- ① void methodA() {}
- ② static void methodA() {}
- ③ int methodA() {}
- ④ final void methodA() {}

7. 다음의 실행의 결과는?

```
클래스 선언부
class Parent{}
class DerivedOne extends Parent{}
class DrivedTwo extends Parent{}

Parent p = new Parent();
DerivedOne d1 = new DerivedOne();
DerivedTwo d2 = new DerivedTwo();
p = d1;
```

- ① 컴파일 에러남
- ② 컴파일은 되나 실행 시 에러남
- ③ 컴파일 실행 모두 잘 됨
- ④ 답 없음

8. 다음 프로그램을 수행하면 어떤 결과가 나오는지 선택하시오.

```
class Exercise {
    public static void main(String[] args) {
        String str = "12345";
        int sum = 0;
        for (int i=0;i<str.length();i++) {
            sum += str.charAt(i) - '0';
        }
        System.out.println("sum="+sum);
    }
}
```

- ① 에러남
- ② 0
- ③ 100
- ④ 15

9. 다음은 배열을 선언하거나 초기화 한 것이다. 잘못된 것을 고르시오.

- ① int[] arr[]
- ② int[] arr = {1,2,3};
- ③ int[] arr = new int[5];
- ④ int[] arr = new int[5]{1,2,3,4,5};

10. 다음과 같은 배열이 있을 때, arr[3].length의 값은 얼마인가?

```
int[][] arr = {
    {5,5,5,5,5},
    {10,10,10},
    {20,20,20,20},
    {30,30}
};
```

- ① 5
- ② 3
- ③ 4
- ④ 2

11. 다음은 거스름돈을 몇 개의 동전으로 지불할 수 있는지를 계산하는 문제이다. 10원은 몇 개를 받을 수 있습니까?

```
// 큰 금액의 동전을 우선적으로 거슬러 줘야한다.
int[] coinUnit = {500, 100, 50, 10};
int money = 2680;
System.out.println("money=" + money);
for (int i=0;i<coinUnit.length;i++) {
    System.out.println(coinUnit[i]+"원"+money/coinUnit[i]);
    money = money%coinUnit[i];
}
```

- ① 5
- ② 1
- ③ 2
- ④ 3

12. 다음의 코드에 정의된 변수들의 갯수가 맞는 것을 찾으시오.

```
class PlayingCard {
    int kind;
    int num;
    static int width;
    static int height;
    PlayingCard(int k, int n) {
        kind = k;
        num = n;
    }
    public static void main(String args[]) {
        PlayingCard card = new PlayingCard(1,1);
    }
}
```

- ① 클래스 변수 : 2, 인스턴스 변수 : 2, 지역변수 : 2
- ② 클래스 변수 : 2, 인스턴스 변수 : 4
- ③ 클래스 변수 : 2, 인스턴스 변수 : 2, 지역변수 : 3
- ④ 클래스 변수 : 2, 지역 변수 : 5

13. 다음 중 생성자에 대한 설명으로 옳지 않은 것을 모두 고르시오.

- ① 모든 생성자의 이름은 클래스의 이름과 동일해야 한다.
- ② 생성자는 객체를 생성하기 위한 것이다.
- ③ 클래스에는 생성자가 반드시 하나 이상 있어야 한다.
- ④ 생성자가 없는 클래스는 컴파일러가 기본 생성자를 추가한다.
- ⑤ 생성자는 오버로딩할 수 없다.

14. 다음 중 this에 대한 설명으로 맞지 않은 것을 모두 고르시오.

- ① 객체 자신을 가리키는 참조변수이다.
- ② 클래스 내에서라면 어디서든 사용할 수 있다.
- ③ 지역변수와 인스턴스변수를 구별할 때 사용한다.
- ④ 클래스 메서드 내에서는 사용할 수 없다.

15. 다음 중 오버로딩이 성립하기 위한 조건이 아닌 것을 모두 고르시오.

- ① 메서드의 이름이 같아야 한다.
- ② 매개변수의 개수나 타입이 달라야 한다.
- ③ 리턴 타입이 달라야 한다.
- ④ 매개변수의 이름이 달라야 한다.

16. 다음 중 아래의 add메서드를 올바르게 오버로딩 한 것을 모두 고르시오.

```
long add(int a, int b) {
    return a + b;
}
```

- ① long add(int x, int y) {return x + y;}
- ② long add(long a, long b) {return a + b;}
- ③ int add(byte a, byte b) {return a + b;}
- ④ int add(long a, int b) {return (int)(a + b);}

17. 다음의 코드를 컴파일하면 컴파일 에러가 발생한다. 컴파일 에러가 발생하지 않는 라인은?

```
class MemberCall{
    int iv = 10;
    static int cv = 20;
    int iv2 = cv;
    static int cv2 = iv;    // 라인 1
    static void staticMethod1() {
        System.out.println(cv);
        System.out.println(iv);    // 라인 2
    }
    void instanceMethod1() {
        System.out.println(cv);
        System.out.println(iv);    // 라인 3
    }
    static void staticMethod2() {
        staticMethod1();
        instanceMethod1();    // 라인 4
    }
}
```

- ① 라인 1
- ② 라인 2
- ③ 라인 3
- ④ 라인 4

18. 중 오버라이딩의 조건으로 옳지 않은 것을 모두 고르시오.

- ① 조상의 메서드와 이름이 같아야 한다.
- ② 매개변수의 수와 타입이 모두 같아야 한다.
- ③ 리턴타입이 같아야 한다.
- ④ 제어자는 조상의 메서드보다 좋은 범위로만 변경할 수 있다.
- ⑤ 조상의 메서드보다 더 많은 수의 예외를 선언할 수 있다.

19. 다음 중 접근제어자를 접근범위가 넓은 것에서 좁은 것의 순으로 바르게 나열할 것은?

- ① public - protected - (default) - private
- ② public - (default) - protected - private
- ③ (default) - public - protected - private
- ④ private - protected - (default) - public

20. 다음 중 연산 결과가 true가 아닌 것을 모두 고르시오.

```
class {}
class FireEngine extends Car implements Movable {}
class Ambulance extends Car {}
FireEngine fe = new FireEngine();
```

- ① fe instanceof FireEngine
- ② fe instanceof Movable
- ③ fe instanceof Object
- ④ fe instanceof Car
- ⑤ fe instanceof Ambulance