1. What is the difference between overload and override?
   a. There is no difference.
   b. In method overloading, methods have the same name and different signatures. In method overriding, the methods must have the same name and same signature.
   c. In method overloading, methods have the different name and same signatures. In method overriding, the methods must have the same name and same signature.
   d. In method overloading, methods have the same name and different signatures. In method overriding, the methods must have the same name and different signature.

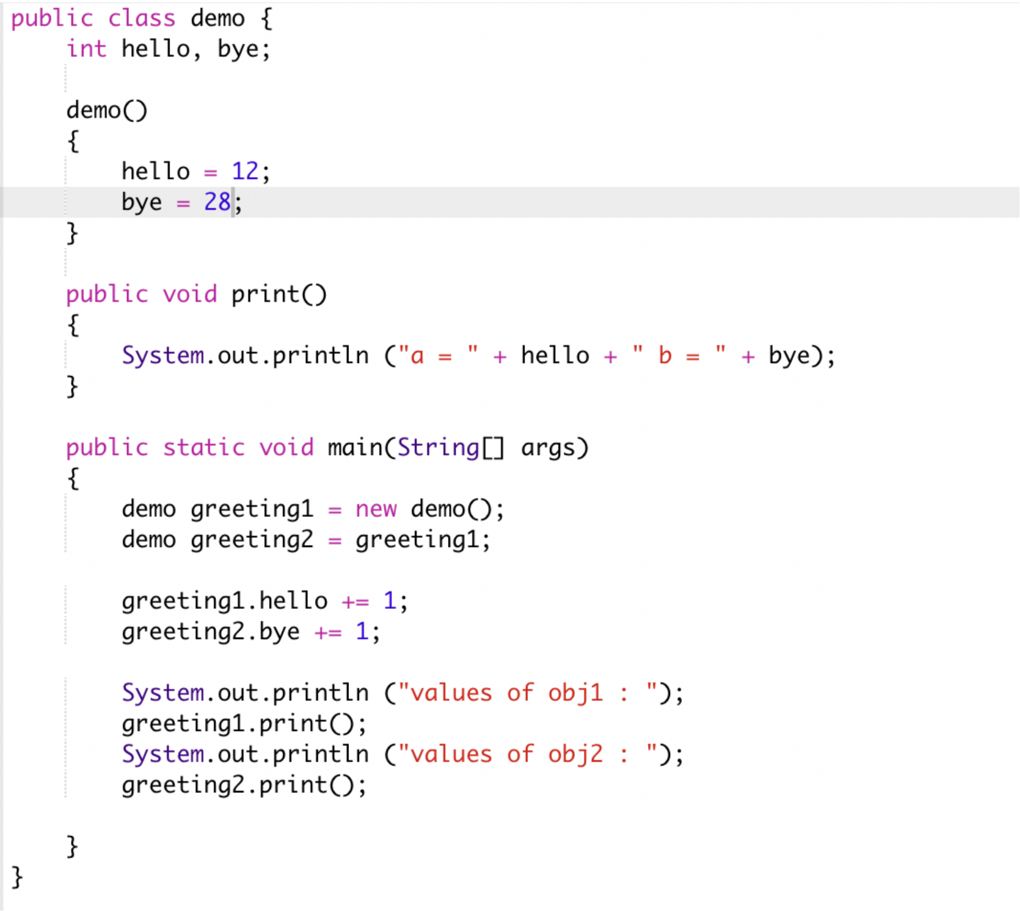
2. What is the Java keyword we can use to check if the object is an instance of a specific type?
   a. Equals
   b. ==
   c. Instanceof
   d. =

3. When we implement an interface method, it must declared as:
   a. Private
   b. Protected
   c. Public

4. Create an `ArrayList` with an initial capacity of 30 references to an `Object`.

   a. `Object array(30) = new ArrayList();`
   b. `ArrayList array[30] = new ArrayList();`
   c. `ArrayList<Object> list = new ArrayList<>(30);`
   d. `ArrayList<reference> list = new Arraylist(reference);`

5. What is the output?



```java
public class demo {
    int hello, bye;

    demo()
    {
        hello = 12;
        bye = 28;
    }

    public void print()
    {
        System.out.println ("a = " + hello + " b = " + bye);
    }

    public static void main(String[] args)
    {
        demo greeting1 = new demo();
        demo greeting2 = greeting1;

        greeting1.hello += 1;
        greeting2.bye += 1;

        System.out.println ("values of obj1 : ");
        greeting1.print();
        System.out.println ("values of obj2 : ");
        greeting2.print();

    }
}
```

a. values of obj1: a =13 b=29 values of obj2: a =12 b=28
b. values of obj1: a =12 b=28 values of obj2: a =13 b=29
c. values of obj1: a =12 b=29 values of obj2: a =12 b=28
d. values of obj1: a =13 b=29 values of obj2: a =13 b=29