# Java Review
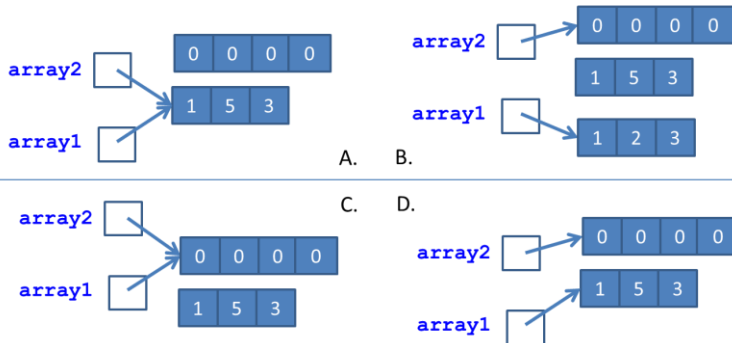
**Find the correct memory model**

```
int[] array1 = {1, 2, 3};
int[] array2 = array1;
array2[1] = 5;
array2 = new int[4];
```



A.       B.

C.       D.

**What is the output of the code snippet below?**

```
int[] nums = {3, 7, 21};
nums = new int[4];
nums[1] = 6;
nums[3] = 42;
for (int n : nums) {
    System.out.print(n + ", ");
}
```

A.0, 6, 0, 42

B.3, 6, 21, 42

C.Error: cannot increase the length of an array

D.3, 7, 21, 42

E.Error: invalid for loop

---

1. Suppose **A** is an interface.
   Can you create an instance using  **new A();**            A. Yes     B. No
   Can you declare a reference var x as:  **A x;**            A. Yes     B. No

2. Suppose **A** is an abstract class.
   Can you create an instance using  **new A();**            A. Yes     B. No
   Can you declare a reference var x as:  **A x;**            A. Yes     B. No

3. It is possible that an interface has a concrete method.        A. True     B. False
4. An abstract class may not have an abstract method.             A. True     B. False
5. An abstract class may implement multiple interfaces.           A. True     B. False
6. An interface can extend from an abstract class.                A. True     B. False
7. An abstract class cannot be a child of another class, whether the parent is abstract or not.
                                                        A. True     B. False
8.  We can have multiple layers of inheritance on interfaces.  A. True     B. False

---

```
public class Student{
   public String name;
   public double grade;
   public Student (String name, double grade){
     this.name = name;
     this.grade = grade;
   }


   //Want to write a compare method to compare
   //two students based on their grade. It returns
   //name of the student with the higher grade
}
```

Should this compare method be static or non-static?
A: It should be static
B: It should be non-static
C: It depends on how you design this method
D: I really have no idea.