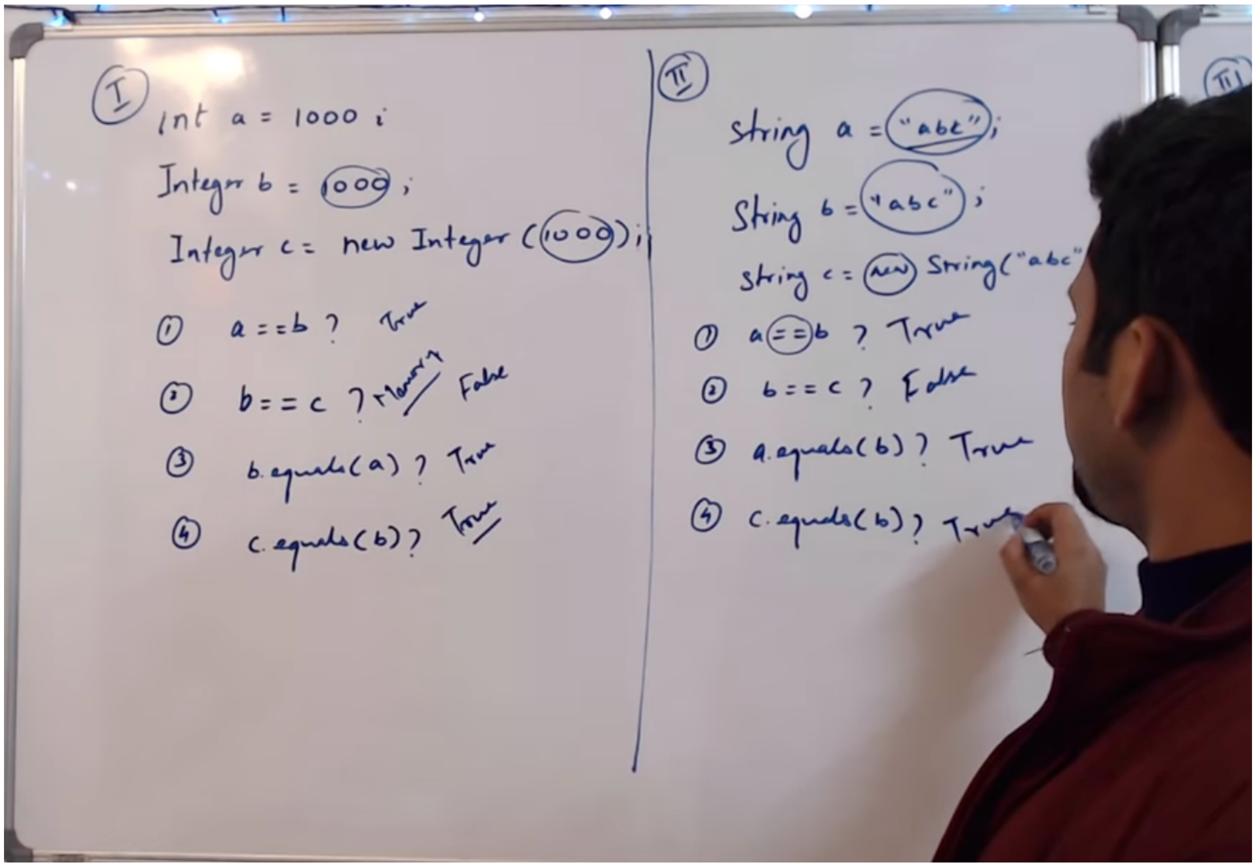


I

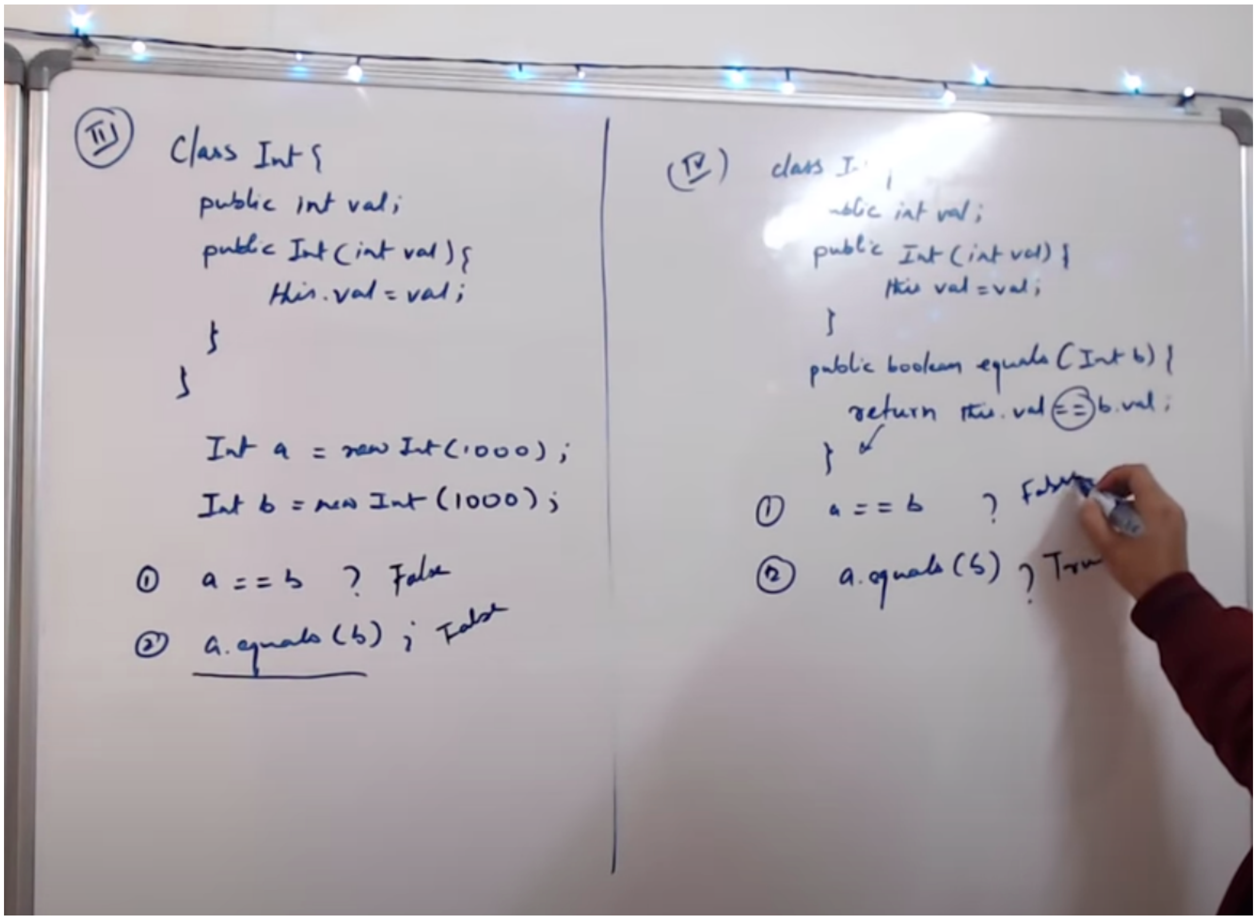
```
int a = 1000 ;
Integer b = 1000 ;
Integer c = new Integer (1000);
```

1. a == b ? True
2. b == c ? [illegible] False
3. b.equals(a) ? True
4. c.equals(b) ? True

II

```
String a = "abc";
String b = "abc" ;
String c = new String("abc"
```

1. a == b ? True
2. b == c ? False
3. a.equals(b) ? True
4. c.equals(b) ? True

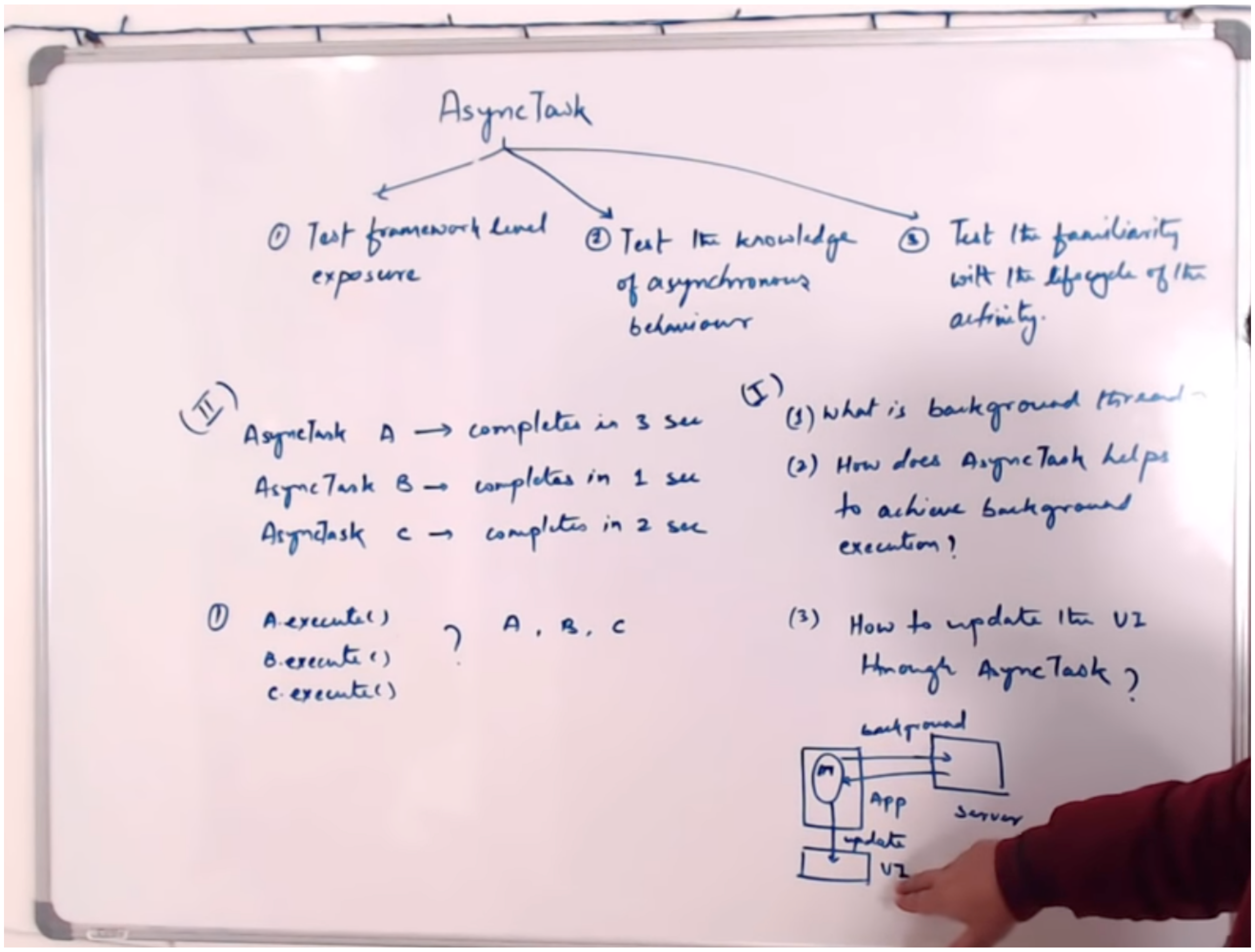


```
(III) Class Int {
        public int val;
        public Int(int val){
            this.val = val;
        }
      }

        Int a = new Int(1000);
        Int b = new Int(1000);

   ① a == b ? False
   ② a.equals(b) ; False
```

```
(IV) class Int {
        public int val;
        public Int(int val) {
            this.val = val;
        }
        public boolean equals(Int b) {
            return this.val == b.val;
        }
   ① a == b ? Fals
   ② a.equals(b) ? Tru
```



AsyncTask

① Test framework level exposure

② Test the knowledge of asynchronous behaviour

③ Test the familiarity with the lifecycle of the activity.

(I)

(1) What is background thread

(2) How does AsyncTask helps to achieve background execution?

(3) How to update the UI through AsyncTask?

background
App
server
update
UI

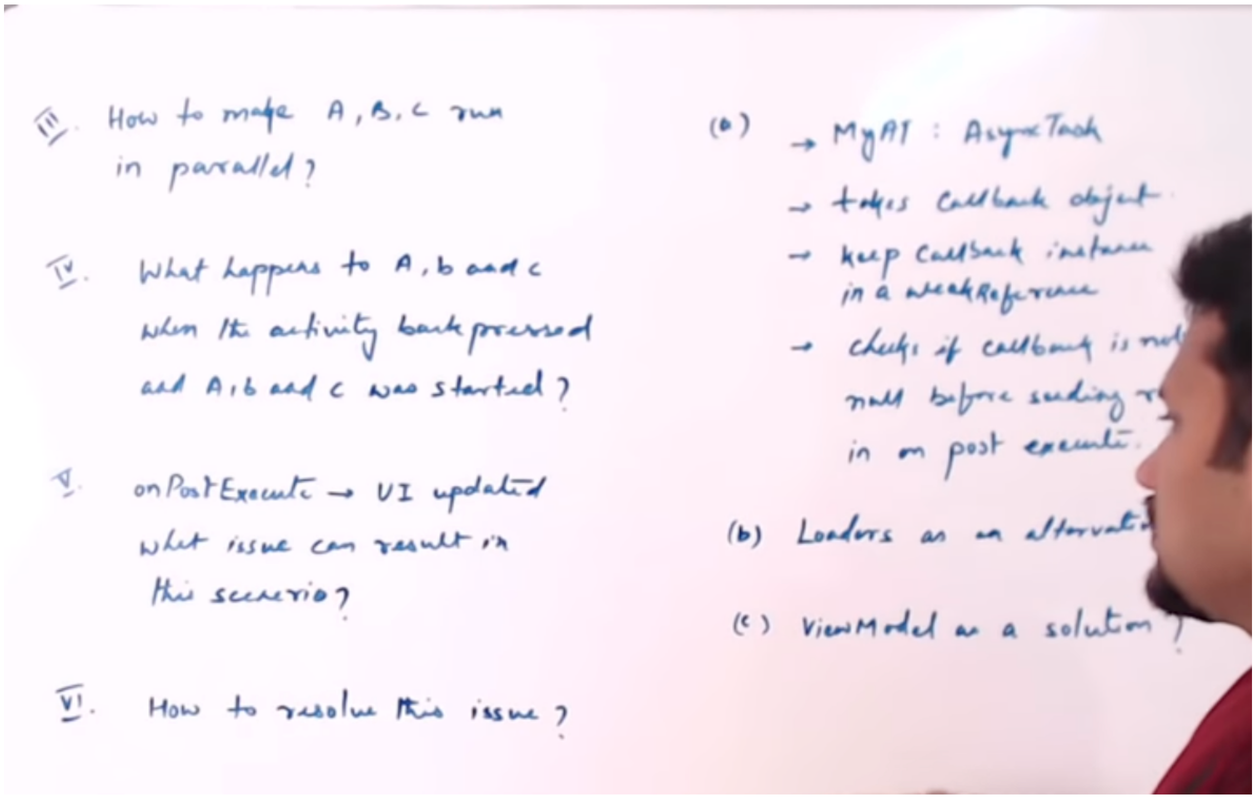
(II)

AsyncTask A → completes in 3 sec

AsyncTask B → completes in 1 sec

AsyncTask C → completes in 2 sec

① A.execute()
B.execute()
C.execute()
? A , B , C



III. How to make A, B, C run in parallel?

IV. What happens to A, b and c when the activity back pressed and A, b and c was started?

V. onPostExecute → UI updated what issue can result in this scenerio?

VI. How to resolve this issue?

(a) → MyAT : AsyncTask

→ takes Callback object.

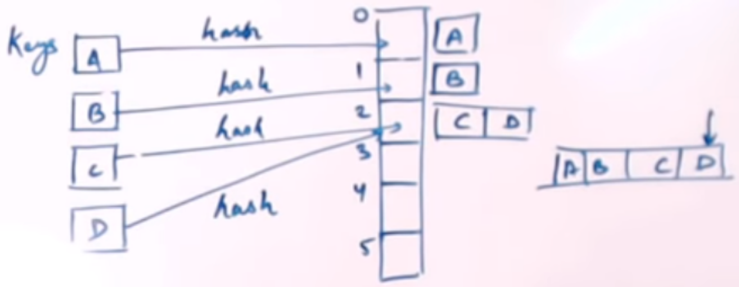
→ keep Callback instance in a weakReference

→ checks if callback is not null before sending re

in on post execute.

(b) Loaders as an alternati

(c) ViewModel as a solution?



O(1)

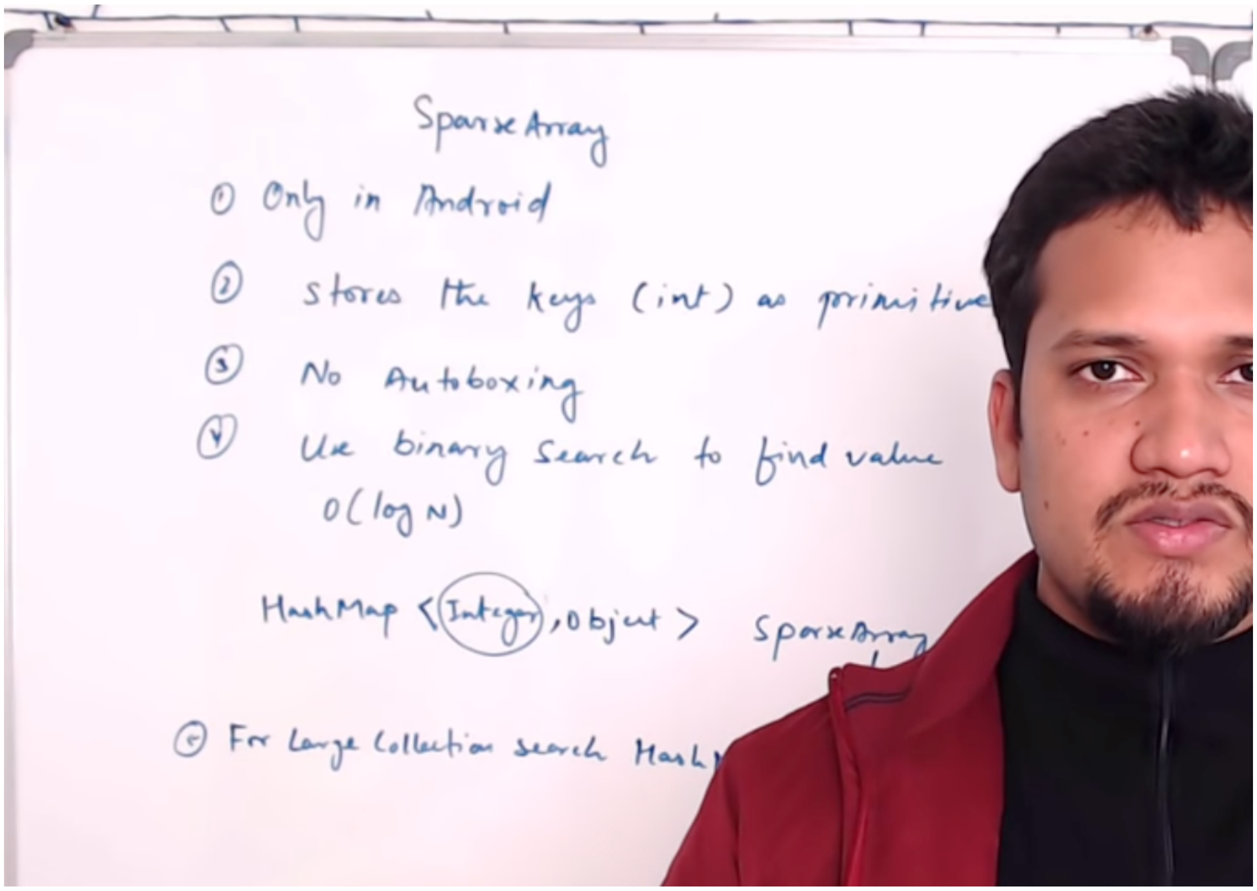
ject) ⟶ Some 32 bit Integer ⟶ index between 0 to M

- - - ⟶ hashCode - - - ⟶ hash

A equals B
⟶ same hashcode

① 31x+y rule
② primitime use wrapper type
③ Null ⟶ 0
④ Array ⟶ Arrays.deepHashCode()

```
int hash = 17
hash = 31 x a.hashCode() + hash
hash = 31 x b.hashCode() + hash
```



Sparse Array

1. Only in Android
2. stores the keys (int) as primitive
3. No Autoboxing
4. Use binary search to find value O(log N)

HashMap <Integer, Object> SparseArray

5. For Large Collection search Hash