

## Interviewer mayur and man

### We have employee object fields

Id

Name

Age

Gender

We have method

Arg- list of employee objects

Example

Id : 1

Name : mayor

Age : 30

Gender : male

Id : 2

Name : aman

Age : 28

Gender : male

Id : 1

Name : syed

Age : 34

Gender : male

Composite keys :

Name

Id

```
List <Employee> employees = new ArrayList();
```

1. Requirement 1: list of uniq employees

Requirement 2: age > 30

```
List<Employee> employess= Employees.stream().filter(e->e.getAge()>30  
&& e.get()<35).collect(Collectors.toList())
```

Emoloyees.stream().

2. Predict the output :

Needs explanation on equals and hascode method

If we implement equals and hashcode properly than size will 1

```
class Person{
```

```
String name;
```

```
Int age;
```

```
String emailId;
```

```
String phn;
```

```
}
```

```
Person p1 = new Person("A",32,"abc@bcd.com","1234")
```

```
Person p2 = new Person("A",32,"abc@bcd.com","1234")
```

```
Set<Person> set = new HashSet<>();
```

```
Set.add(p1);
```

```
Set.add(p2);
```

```
System.out.println(set.size());
```

3. Write Rest service using spring boot java with service and repository

```
@RestController
```

```
@RequestMapping("/myapi")
```

```
Class MyController {
```

```
@Autorwred
```

```
MuyServiceImpl my service
```

```
@Get("/getObj")
```

```
Public ResponseEntity<String> mygetAPICall(){
```

```
Return ReesponseEntity.status(HttpStatus.Ok).body("Hello");
```

```
}
```

```
}
```

```
@Repository
EmployeeRepository extends JpaRepository<Employee, Long> {

}
```

```
Interface MyService{
Public ge

}
```

```
@Service
Package com.myapi.myserver
Class MyServiceImpl implements MyService {
@Autowired
EmployeeRepository empRepo;
```

```
@Override
Public Employee getAllEmployees(){
Return. empRepo.getAll();

}
```

```
}
```

Predict the output

```
public class Hello {

    int a;
    private static int b;

    static {
        System.out.println("value of b inside static block="+b);
    }
    {
        System.out.println("value of b inside instance block =" +b);
    }

}
```

```

public Hello() {

    System.out.println("value of a inside constructor =" + a);
}

public void test () {
    System.out.println("value of a inside test=" + a);
}

public static void main(String args[]) {

    Hello hello = new Hello();
    hello.test();
}
}

```

O/p :

value of b inside static block= 0  
 value of b inside instance block =0  
 value of a inside constructor =0  
 value of a inside test=0

4. What is difference between interceptors and filters.
5. Scenario based, if there is outbound call to update field and same field needed to be updated at local database end, how to handle the situation to maintain consistency of field if any one REST call or db call failed to update.

Ans: First call db to update the value and before committing db call we need to call REST api out bound call, depends on response from API call decide weather we need to proceed with committing the db call