Java Stream API Interview Practice Problems (with Solutions)

1. Find Employees whose names start with 'S' and salary > 50000

```
------------------------------------------------------------
List<Employee> result = employees.stream()
    .filter(e -> e.getName().startsWith("S") && e.getSalary() > 50000)
    .collect(Collectors.toList());

------------------------------------------------------------
```

2. Find Duplicate Employee Names

```
------------------------------------------------------------
List<String> duplicates = employees.stream()
    .collect(Collectors.groupingBy(Employee::getName, Collectors.counting()))
    .entrySet().stream()
    .filter(e -> e.getValue() > 1)
    .map(Map.Entry::getKey)
    .collect(Collectors.toList());

------------------------------------------------------------
```

3. Get a Comma-Separated String of All Employee Names in Uppercase

```
------------------------------------------------------------
String names = employees.stream()
    .map(e -> e.getName().toUpperCase())
    .collect(Collectors.joining(", "));

------------------------------------------------------------
```

4. Find Total Salary of Employees in 'HR' Department

```
------------------------------------------------------------
```

```
double totalSalary = employees.stream()
    .filter(e -> e.getDepartment().equalsIgnoreCase("HR"))
    .mapToDouble(Employee::getSalary)
    .sum();
```

----

5. Group Employees by Department and Count Each Department

```
Map<String, Long> result = employees.stream()
    .collect(Collectors.groupingBy(Employee::getDepartment, Collectors.counting()));
```

----

6. Group Employees by Department and Find Total Salary

```
Map<String, Double> totalSalaryByDept = employees.stream()
    .collect(Collectors.groupingBy(
        Employee::getDepartment,
        Collectors.summingDouble(Employee::getSalary)
    ));
```

----

7. Find the Highest-Paid Employee in Each Department

```
Map<String, Optional<Employee>> highestPaidByDept = employees.stream()
    .collect(Collectors.groupingBy(
        Employee::getDepartment,
        Collectors.maxBy(Comparator.comparing(Employee::getSalary))
    ));
```
