Java + Spring Boot Coding Questions with Answers (2 Years Experience)

1. Reverse a String

Write a method to reverse a string without using library methods.

```
public String reverseString(String input) {
    StringBuilder result = new StringBuilder();
    for (int i = input.length() - 1; i >= 0; i--) {
        result.append(input.charAt(i));
    }
    return result.toString();
}
```

2. Check if String is Palindrome

Check whether a given string is a palindrome.

```
public boolean isPalindrome(String str) {
    int left = 0, right = str.length() - 1;
    while (left < right) {
        if (str.charAt(left++) != str.charAt(right--)) return false;
    }
    return true;
}
```

3. Count Character Occurrences

```
public Map<Character, Integer> countCharacters(String str) {
    Map<Character, Integer> map = new HashMap<>();
    for (char c : str.toCharArray()) {
        map.put(c, map.getOrDefault(c, 0) + 1);
    }
    return map;
}
```

4. First Non-Repeating Character

```
public char firstNonRepeatingChar(String s) {
    Map<Character, Integer> map = new LinkedHashMap<>();
    for (char c : s.toCharArray()) {
        map.put(c, map.getOrDefault(c, 0) + 1);
    }
    for (Map.Entry<Character, Integer> entry : map.entrySet()) {
        if (entry.getValue() == 1) return entry.getKey();
    }
    return '_';
}
```

5. Remove Duplicates from List

```
public List<Integer> removeDuplicates(List<Integer> list) {
    return new ArrayList<>(new HashSet<>(list));
}
```

6. Second Highest in Array

```
public int secondHighest(int[] arr) {
    int first = Integer.MIN_VALUE, second = Integer.MIN_VALUE;
    for (int num : arr) {
        if (num > first) {
            second = first;
            first = num;
        } else if (num > second && num != first) {
            second = num;
        }
    }
    return second;
}
```

7. Check for Anagram

```
public boolean isAnagram(String s1, String s2) {
    if (s1.length() != s2.length()) return false;
    char[] a1 = s1.toCharArray(), a2 = s2.toCharArray();
    Arrays.sort(a1); Arrays.sort(a2);
    return Arrays.equals(a1, a2);
}
```

8. Stream API - Filter Even Numbers

```
List<Integer> evens = list.stream()
```

```
    .filter(n -> n % 2 == 0)
    .collect(Collectors.toList());
```

9. Sum of Squares using Stream API

```
int sum = list.stream()
    .map(x -> x * x)
    .reduce(0, Integer::sum);
```

10. Group Employees by Department

```
Map<String, List<Employee>> grouped = employees.stream()
    .collect(Collectors.groupingBy(Employee::getDepartment));
```

11. Spring Boot - Get Employees with Salary > 50000

```
@GetMapping("/high-salary-employees")
public List<String> getHighSalaryEmployees() {
    return employeeRepository.findAll().stream()
        .filter(emp -> emp.getSalary() > 50000)
        .map(Employee::getName)
        .collect(Collectors.toList());
}
```