

4 Object-Oriented Programming (rkh23)

(a) Consider the following class definition.

```
public class User {
    private String username;
    public User(String uname) { username = uname; }
    public void setUsername(String uname) {username = uname; }
    @Override public boolean equals(Object o) { ... }
    @Override public int hashCode() { ... }
}
```

You may assume `equals()` and `hashCode()` are correctly implemented and both refer to `username`.

(i) Explain the contract between `equals()` and `hashCode()` and provide an example of why it is necessary. [3 marks]

(ii) Identify a critical flaw in the `User` class design regarding its use as a `Map` key. Illustrate your answer with an example of the flaw. [3 marks]

(b) Consider the following Java code snippet.

```
List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);
System.out.println("Start");
Integer result =
    numbers.stream()
        .filter(n -> {
            System.out.print("Filter(" + n + ") ");
            return n % 2 == 0; })
        .map(n -> {
            System.out.print("Map(" + n + ") ");
            return n * 10; })
        .findFirst()
        .orElse(0);
System.out.println("\nResult: " + result);
```

(i) Explain what it means for a Java Stream to be lazy. Distinguish between intermediate and terminal operations, with examples, and explain how laziness affects their execution. [6 marks]

(ii) The `findFirst()` method returns an `Optional<Integer>` rather than a raw `Integer`. Explain the purpose of the `Optional` class and why `orElse(0)` is used in this snippet. [2 marks]

(iii) Provide a detailed explanation of the code execution and state the exact text printed to the console when this code runs. [6 marks]