

# RandomPercentage Assignment

---

## Overview

This assignment focuses on using Java's `Random` class to generate random numbers and format them as percentages using either `printf` or `DecimalFormat`.

## Learning Objectives

- Understand the difference between seeded and unseeded `Random` objects
- Generate random double values using `nextDouble()`
- Format numbers to specific decimal places
- Convert decimal values to percentages
- Use either `printf` or `DecimalFormat` for number formatting

## Assignment Requirements

### Task: Random Percentage Generator

Create a Java program that demonstrates the use of `Random` objects and number formatting.

#### Requirements:

##### 1. Create Two Random Objects:

- One `Random` object without a seed (unseeded)
- One `Random` object with any seed value (seeded)

##### 2. Generate Random Values:

- Generate random double values from both `Random` objects
- Use `nextDouble()` method to get values between 0.0 and 1.0

##### 3. Format as Percentages:

- Format the values to exactly 2 decimal places
- Use either `printf` format specifier OR `DecimalFormat` with the correct pattern
- Display with `%` symbol

##### 4. Output Format:

- Display both unseeded and seeded random values
- Each value should be formatted as below

## Example Output

```
Unseeded Random: 45.67%
Seeded Random: 23.45%
```

## Key Concepts

### Random Class

- `Random()` - Creates unseeded random number generator
- `nextDouble()` - Returns random double between 0.0 and 1.0

### Seeded vs Unseeded Behavior

- **Unseeded Random:** Produces different values each time program runs
- **Seeded Random:** Produces same sequence of values each time (deterministic)

## Test Cases

The autograder tests the following:

1. **Program Compilation:** Code compiles without errors
2. **Random Generation:** Program generates random values
3. **Percentage Formatting:** Values are displayed as percentages with % symbol
4. **Decimal Precision:** Values are formatted to exactly 2 decimal places
5. **Value Range:** Percentages are in valid 0-100% range
6. **Both Objects Used:** Both unseeded and seeded random values are displayed
7. **Seeded Consistency:** Seeded random produces consistent values across runs

## Tips

- Seeded random will always produce the same values for the same seed
- Unseeded random will produce different values each time you run the program