

Java Coding Challenges - Part 7: Date, Time, and Numeric Objects

1. Challenge: Get current date and time using LocalDateTime

```
import java.time.LocalDateTime;

public class CurrentDateTime {
    public static void main(String[] args) {
        LocalDateTime now = LocalDateTime.now();
        System.out.println("Current Date and Time: " + now);
    }
}
```

2. Challenge: Calculate age given a birth date

```
import java.time.LocalDate;
import java.time.Period;

public class CalculateAge {
    public static void main(String[] args) {
        LocalDate birthDate = LocalDate.of(2000, 1, 1);
        LocalDate today = LocalDate.now();
        Period age = Period.between(birthDate, today);
        System.out.println("Age: " + age.getYears() + " years");
    }
}
```

3. Challenge: Format date in dd-MM-yyyy format

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;

public class FormatDate {
    public static void main(String[] args) {
        LocalDate date = LocalDate.now();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy");
        System.out.println("Formatted Date: " + date.format(formatter));
    }
}
```

4. Challenge: Add 5 days to current date and print

```
import java.time.LocalDate;

public class AddDays {
    public static void main(String[] args) {
```

```
        LocalDate today = LocalDate.now();
        LocalDate futureDate = today.plusDays(5);
        System.out.println("Date after 5 days: " + futureDate);
    }
}
```

5. Challenge: Round a decimal to 2 places using BigDecimal

```
import java.math.BigDecimal;
import java.math.RoundingMode;

public class RoundDecimal {
    public static void main(String[] args) {
        BigDecimal number = new BigDecimal("123.456789");
        BigDecimal rounded = number.setScale(2, RoundingMode.HALF_UP);
        System.out.println("Rounded: " + rounded);
    }
}
```