DAY 11

1. What is a Decision-Making Statement?

A **decision-making statement** is a control flow tool that lets your program execute a specific block of code based on a condition. It's the core logic that allows programs to make choices, such as checking if a user's password is correct before granting access.

---

2. The switch Statement in Java

The switch statement is a powerful alternative to a long chain of if-else if-else statements. It's designed for situations where you need to compare a single variable against multiple, distinct values.

Real-time use case:

Think of a menu in a command-line application. The program can take a user's input (e.g., '1', '2', or '3') and use a switch statement to execute the corresponding action, such as "View Balance," "Withdraw Funds," or "Exit."

---

3. Syntax and Hands-on Experience with switch

The syntax for a switch statement is straightforward. The switch expression is evaluated once, and its value is compared against each case label.

Java

```
public class SwitchExample {
   public static void main(String[] args) {
      int day = 3;
      String dayName;

      switch (day) {
         case 1:
            dayName = "Sunday";
            break;
         case 2:
            dayName = "Monday";
            break;
         case 3:
            dayName = "Tuesday";
            break;
```

```
            case 4:
                dayName = "Wednesday";
                break;
            case 5:
                dayName = "Thursday";
                break;
            case 6:
                dayName = "Friday";
                break;
            case 7:
                dayName = "Saturday";
                break;
            default:
                dayName = "Invalid Day";
                break;
        }
        System.out.println("Today is " + dayName); // Output: Today is Tuesday
    }
}
```

- **switch**: The keyword that initiates the statement.
- **case**: Each case is a label for a specific value. The code block following a matching case is executed.
- **default**: (Optional) The default block is executed if none of the case values match the switch expression. It acts like the else in an if-else statement.

---

4. The break Keyword and Why We Use It

The **break** keyword is essential inside a switch statement. When Java encounters a break, it immediately exits the switch block. Without a break, the program would continue to execute the code for the next case statement, even if the label doesn't match. This is known as **fall-through**.

Real-time use case:

In the bank menu example, after a user selects "View Balance," you want the program to display the balance and then stop. Without a break, it would display the balance and then immediately proceed to the "Withdraw Funds" option, which is not the desired behavior.

5. Interview Questions

1. **What is the purpose of a switch statement, and when would you use it instead of an if-else if-else chain?**
    - **Expected Answer:** Use switch when you need to compare a single variable against multiple, constant values. It often results in cleaner and more readable code than a long if-else if-else chain.
2. **Explain the role of the break keyword in a switch statement.**
    - **Expected Answer:** The break keyword is used to exit the switch statement after a case block has been executed. Without it, the code will "fall through" and execute the next case block as well.
3. **What happens if a switch statement does not have a default case, and no case matches the expression?**
    - **Expected Answer:** The switch statement is simply skipped, and the program continues to execute the code after the switch block. No error will be thrown.
4. **What data types can be used in a switch statement in Java?**
    - **Expected Answer:** The switch expression can be an int, byte, short, char, String, or an enum. Note that long is not supported.
5. **What is "fall-through" in a switch statement?**
    - **Expected Answer:** Fall-through is the behavior where the program continues to execute the code of the next case block after a match has been found, because there is no break statement.