DAY 14

1. What is a Looping Statement?

A **looping statement**, or simply a loop, is a control flow statement that allows you to repeatedly execute a block of code as long as a certain condition remains true. This is fundamental for tasks like iterating through a list of items, performing calculations until a specific value is reached, or processing user input.

- **Real-time use case:** In a game, a loop is used to continuously check for user input and update the game's state (e.g., character position, score). Without a loop, the game would only run once and then quit.

---

2. Different Types of Looping Statements in Java

Java offers several types of loops. While we'll focus on while and do-while today, it's important to know the others as well:

- **for loop:** Used when you know exactly how many times you need to repeat a task.
- **for-each loop:** A simplified loop for iterating over arrays and collections.
- **while loop:** Executes a block of code as long as a condition is true.
- **do-while loop:** Similar to a while loop, but it guarantees that the code block is executed at least once before the condition is checked.

---

3. The while and do-while Statements

The primary difference between these two loops lies in when the condition is checked.

The while Loop

A while loop is an **entry-controlled loop**. The condition is checked at the beginning of the loop. If the condition is true, the code inside the loop is executed. If it's false from the start, the code block is never executed.

- **Syntax:**

Java

```
while (condition) {
   // Code to be executed
}
```

K VENKATA NAGARJUNA REDDY

- **Example:**

Java

```
int count = 1;
while (count <= 5) {
   System.out.println("The count is: " + count);
   count++;
}
// Output:
// The count is: 1
// The count is: 2
// The count is: 3
// The count is: 4
// The count is: 5
```

The do-while Loop

A do-while loop is an **exit-controlled loop**. The code block is executed at least once, and then the condition is checked at the end. This is perfect for situations where you need to perform an action at least once, regardless of the condition.

- **Syntax:**

Java

```
do {
   // Code to be executed
} while (condition);
```

- **Example:**

Java

```
int number = 10;
do {
   System.out.println("The number is: " + number);
   number++;
} while (number < 5);
```

K VENKATA NAGARJUNA REDDY

```
// Output:
// The number is: 10
// The loop runs once, then the condition (11 < 5) is checked and found to be false.
```

---

4. Hands-on Experience with while and do-while

Here's how these loops are used in a practical context.

- **while loop for user input validation:** You can use a while loop to repeatedly prompt a user for input until they provide a valid response.

Java

```
Scanner scanner = new Scanner(System.in);
String password = "";

while (!password.equals("secret123")) {
   System.out.println("Enter the password:");
   password = scanner.nextLine();
}
System.out.println("Access granted!");
```

- **do-while loop for a menu:** You can use a do-while loop to display a menu and get user input at least once before checking if they want to exit.

Java

```
int choice;
do {
   System.out.println("1. Play Game");
   System.out.println("2. View Scores");
   System.out.println("3. Exit");
   System.out.print("Enter your choice: ");
   choice = scanner.nextInt();
} while (choice != 3);
System.out.println("Goodbye!");
```

K VENKATA NAGARJUNA REDDY

DAY 14

5. Interview Questions on Loops

1. **What is a looping statement, and why are loops essential in programming?**
    - **Answer:** A looping statement allows you to execute a block of code repeatedly. They are essential for tasks that require repetition, making your code more efficient and concise.
2. **What is the primary difference between a while loop and a do-while loop?**
    - **Answer:** A while loop checks the condition *before* executing the code block. A do-while loop executes the code block *at least once* before checking the condition.
3. **When would you choose a do-while loop over a while loop?**
    - **Answer:** You would choose a do-while loop when you need to guarantee that the code inside the loop is executed at least one time, regardless of the initial condition. A common use case is for a menu where the user is presented with choices before the exit condition is checked.
4. **What is an infinite loop? How can you create one, and how can you avoid it?**
    - **Answer:** An infinite loop is a loop whose condition never becomes false, causing it to run forever. You can create one with a condition like while (true). You can avoid it by ensuring that the loop's condition will eventually be met, typically by updating a counter or a flag variable inside the loop.
5. **Write a while loop that prints all even numbers from 2 to 10.**
    - **Answer:**

Java

```
int number = 2;

while (number <= 10) {
   System.out.println(number);
   number += 2;
}
```

KVENKATA NAGARJUNA REDDY