1.Java conditional / Control statements with example

A control statement in java is **a statement that determines whether the other statements will be executed or not**. It controls the flow of a program. An 'if' statement in java determines the sequence of execution between a set of two statements. The most commonly used conditional statements are if, if-else, if-else-if, and switch.

1.if Statement : The if statement evaluates a condition; if the condition is true, the code block inside the if statement is executed.

Example:

```
public class Main {
    public static void main(String[] args) {
        int number = 10;


        // Check if number is greater than 0
        if (number > 0) {
            System.out.println("The number is positive.");
        }
    }
}
```

2. if-else Statement : The if-else statement executes one block of code if the condition is true, and another block of code if the condition is false.

Example:

```
public class Main {
    public static void main(String[] args) {
        int number = -10;


        // Check if number is positive or negative
        if (number > 0) {
            System.out.println("The number is positive.");
        } else {
            System.out.println("The number is negative.");
        }
    }
}
```

3. if-else-if Ladder : The if-else-if ladder allows you to check multiple conditions. If one condition is true, the corresponding block of code is executed.

Example:

```
public class Main {
    public static void main(String[] args) {
        int number = 0;

        // Check if number is positive, negative or zero
        if (number > 0) {
            System.out.println("The number is positive.");
        } else if (number < 0) {
            System.out.println("The number is negative.");
        } else {
            System.out.println("The number is zero.");
        }
    }
}
```

4. switch Statement: The switch statement allows you to choose from multiple options based on the value of an expression.

Example:

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

        // Find the name of the day
        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("The day is: " + dayName);
    }
}
```

2. Java conditional / Control loops with example

Control loops in Java are constructs that allow repeated execution of a block of code as long as a specified condition is met. The primary loop types in Java are for, while, and do-while.

1. **for Loop :** The for loop is used when you know in advance how many times you want to execute a statement or a block of statements.

Example:

```
public class Main {
    public static void main(String[] args) {
        // Print numbers from 1 to 5
        for (int i = 1; i <= 5; i++) {
            System.out.println(i);
        }
    }
}
```

2. **while Loop :** The while loop is used when you want to execute a statement or a block of statements as long as a condition is true. The condition is evaluated before executing the loop's body.

Example:

```
public class Main {
   public static void main(String[] args) {
      int i = 1;

      // Print numbers from 1 to 5
      while (i <= 5) {
         System.out.println(i);
         i++;
      }
   }
}
```

3. **do-while Loop :** The do-while loop is similar to the while loop, but the condition is evaluated after executing the loop's body. This guarantees that the loop's body is executed at least once.

   Example:

```
public class Main {
   public static void main(String[] args) {
      int i = 1;

      // Print numbers from 1 to 5
      do {
         System.out.println(i);
         i++;
      } while (i <= 5);
   }
}
```

4. **Enhanced for Loop (for-each Loop) :** The enhanced for loop, also known as the for-each loop, is used to iterate through the elements of an array or a collection.

   Example :

```
public class Main {
   public static void main(String[] args) {
      int[] numbers = {1, 2, 3, 4, 5};

      // Print each number in the array
      for (int number : numbers) {
         System.out.println(number);
```

```
        }
    }
}
```

3. Jump statements and examples

In Java, "jump statements" refer to control flow statements that allow you to transfer control to another part of the program. There are three main jump statements in Java: break, continue, and return.

1. **Break Statement**: The break statement is used to terminate the loop or switch statement it is in and transfer control to the statement immediately following the loop or switch.

   Example:

   ```
   for (int i = 1; i <= 10; i++) {
       if (i == 5) {
           break; // Terminate the loop if i is 5
       }
       System.out.println(i);
   }
   ```

2.Continue Statement: The continue statement is used to skip the current iteration of a loop and continue with the next iteration.

Example:

```
for (int i = 1; i <= 5; i++) {
    if (i == 3) {
        continue; // Skip the current iteration if i is 3
    }
    System.out.println(i);
}
```

3. Return Statement: The return statement is used to exit from a method, optionally passing a value back to the caller.

Example:

```
public int add(int a, int b) {
    int result = a + b;
    return result; // Return the result of addition
}
```