While Loop

What is a While Loop?

A 'While loop' in Java is a control flow statement that allows code to be executed repeatedly based on a given boolean condition. It tests a condition at the start of each loop iteration; if the condition is true, the loop continues to execute.

Structure of a While Loop

Here is the basic syntax structure of a While loop in Java:

```
while (condition) (
// code block to be executed
}
```

condition: This is a boolean expression. As long as this condition evaluates to true, the code block will keep executing.

code block: The set of statements that you want to execute repeatedly.

Usage of While Loop

While loops are particularly useful when the number of iterations is not known beforehand. They are often used in scenarios where you need to repeatedly execute a block of code as long as a certain condition remains true.

Example of a While Loop

To illustrate, here is a simple example that prints numbers from 1 to 5:

```
int i=1;

while (i <= 5) (
System.out.println(i);
 I++;
}
```

- Initialization: Here, int i = 1; initializes the counter.

- Condition: (1 <= 5) checks whether the loop should continue executing.

- Increment: 1++ updates the counter after each loop iteration.

Key Points to Remember

- Ensure that the loop eventually terminates by modifying variables involved in the condition.

- Infinite loops can occur if the condition always evaluates to true and is not correctly managed.

- While loops are versatile and powerful for dynamic iteration needs.

Understanding the 'While loop' is essential as it lays the foundation for controlling program flow in dynamic environments. Practice writing and executing your examples to reinforce these concepts in your programming journey.