Loops

- Loops are used to repeat a block of code.

For Loop

- It is used to run a block of code for a certain number of times.
- It is generally used when we know how many times the loop will iterate.

Syntax

```
for(inititalization;condition;update){
        //body
}
```

- **initialization**: It initialized and/or declares variables and ***executes only once***.
- **condition**: It is evaluated. If the condition is 'true', the body of the for loop is executed.
- **update**: It updates(either increase or decrease) the value of the initial expression.
- The condition is evaluated again and again and the process continues until the condition is false.

Flowchart

Example

1. Display numbers from 1 to 5

CODE

```
package loops;

public class NumbersFor {
    public static void main(String[] args) {
        int n = 5;
        for(int i=1;i<=n;i++){
            System.out.println(i);
        }
    }
}
```

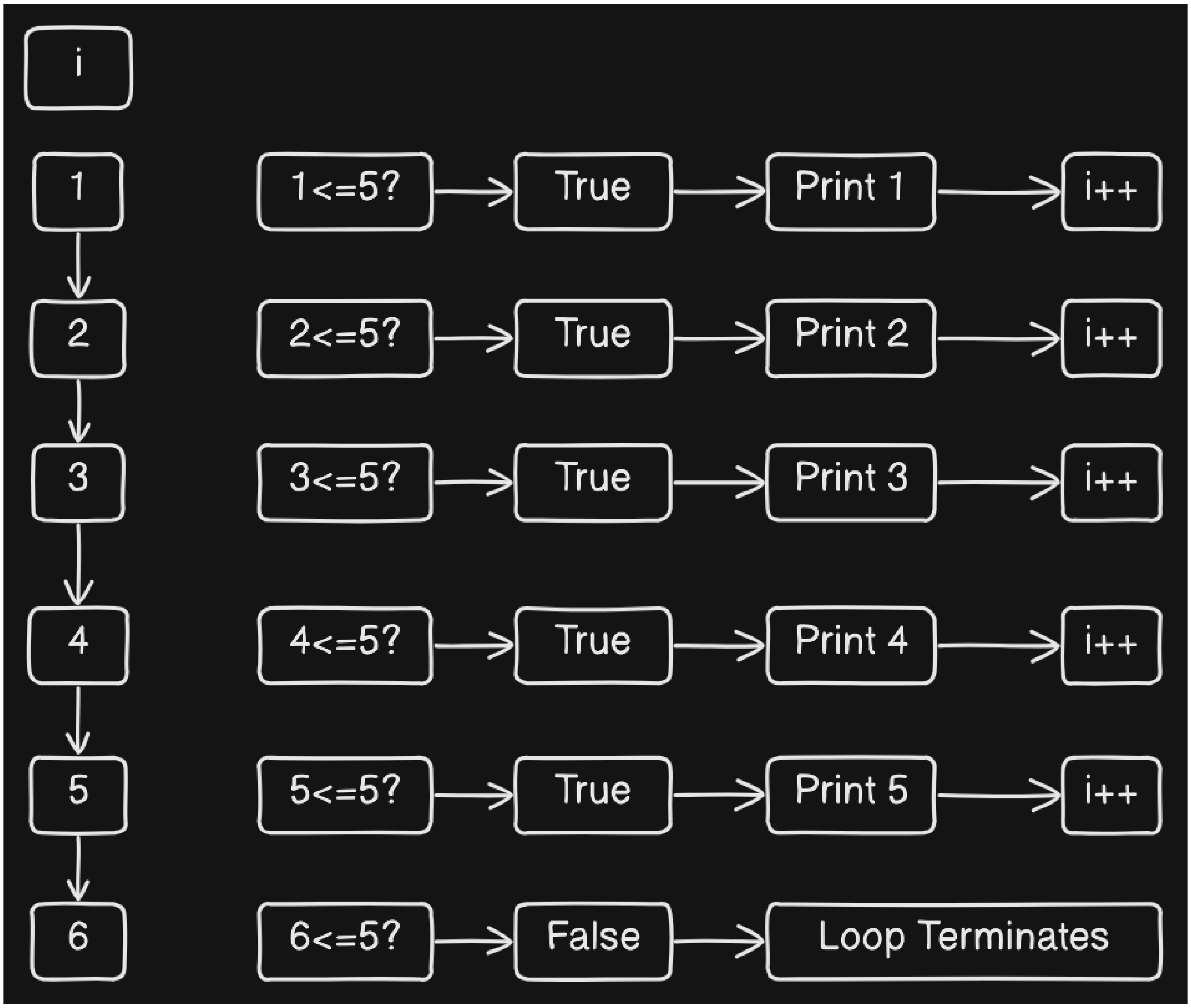
OUTPUT

```
1
2
3
```

```
4
5
```

EXPLANATION

- This loop initializes `i` at 1 and continues while `i` is less than or equal to n(in this case 5). The loop executes 5 times, printing "Batman" each time.



2. Display your name 5 times - 1

CODE

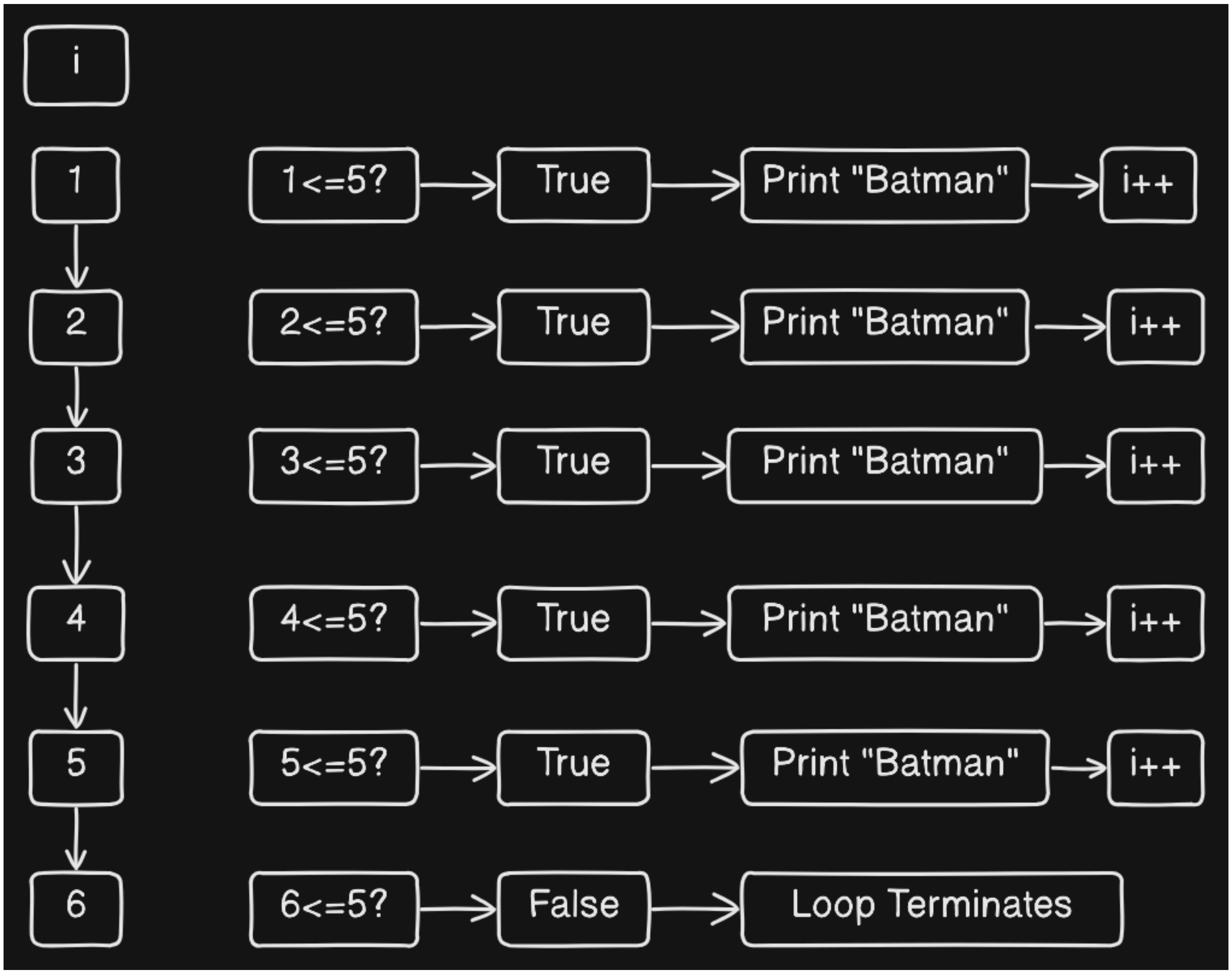
```
package loops;

public class NumbersFor {
    public static void main(String[] args) {
        for(int i=1;i<=5;i++){
            System.out.println("Batman");
        }
    }
}
```

OUTPUT

```
Batman
Batman
Batman
Batman
Batman
```

EXPLANATION

- This loop initializes `i` at 1 and continues while `i` is less than or equal to 5. The loop executes 5 times, printing "Batman" each time.



3. Display your name 5 times - 2

CODE

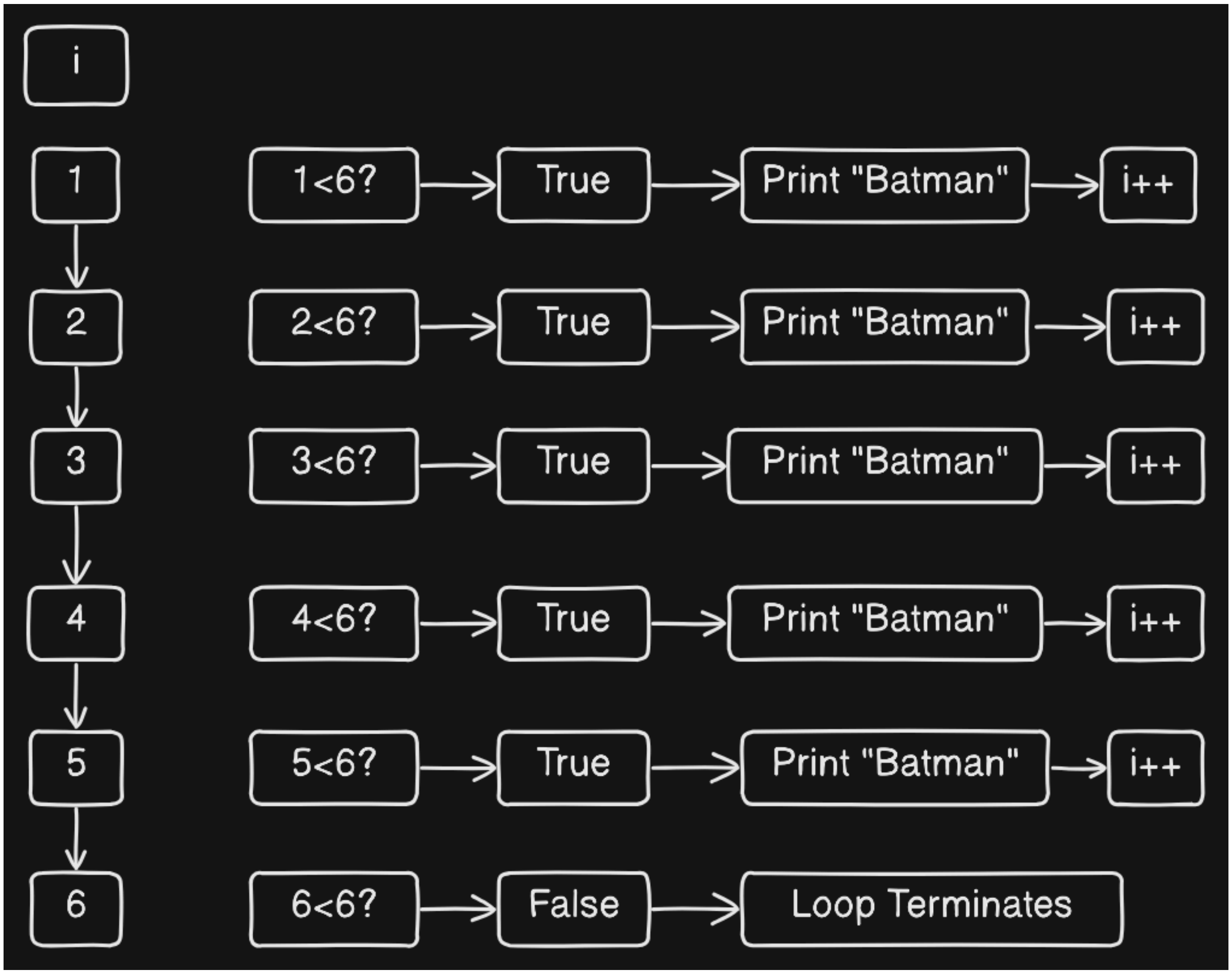
```
package loops;

public class NumbersFor {
    public static void main(String[] args) {
        for(int i=1;i<6;i++){
            System.out.println("Batman");
        }
    }
}
```

OUTPUT

```
Batman
Batman
Batman
Batman
Batman
```

EXPLANATION

- This loop works similarly to the first one but uses `i<6` instead of `i<=5`. Both conditions effectively limit the loop to 5 iterations.



4. Display your name 5 times - 3

CODE

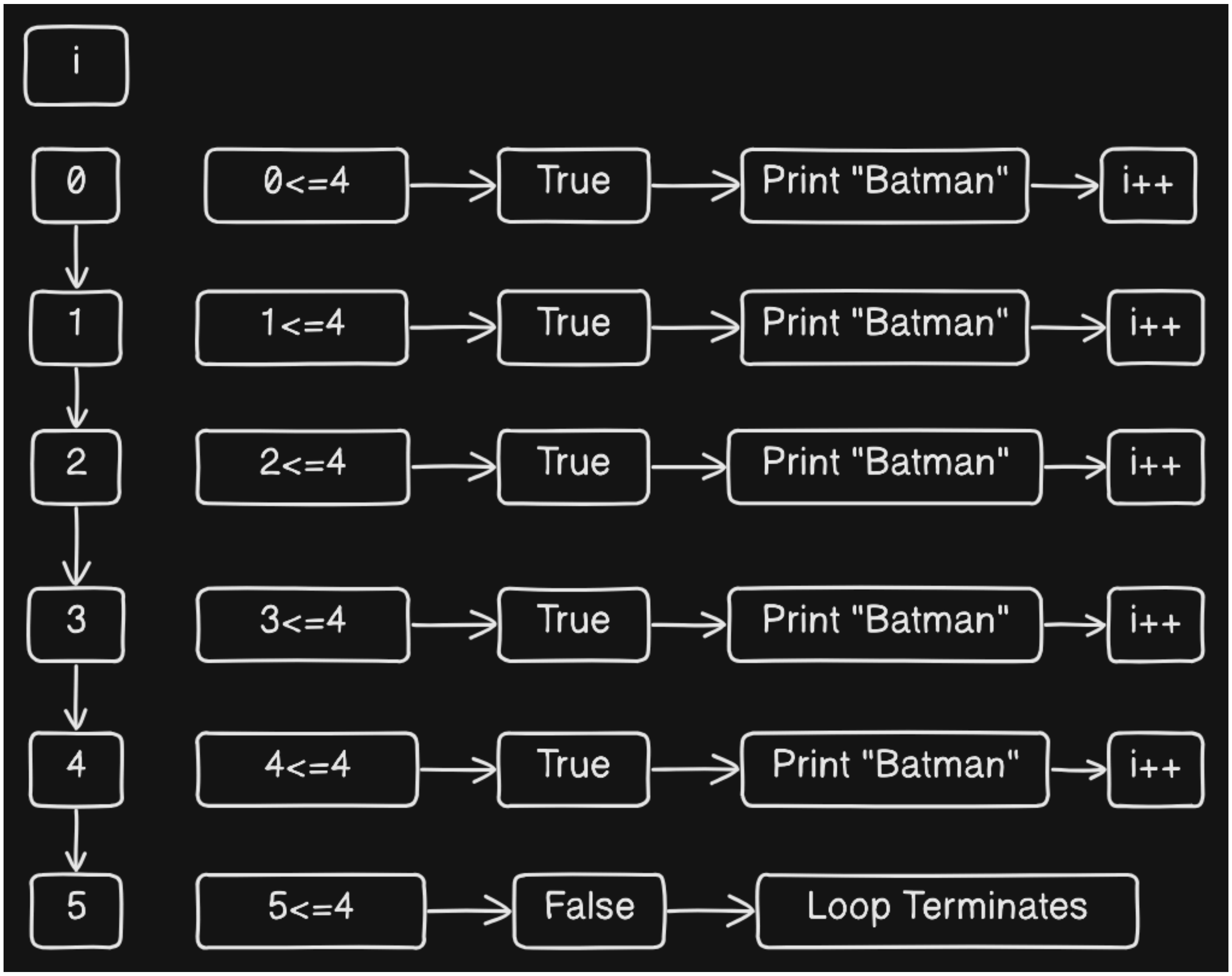
```
package loops;

public class NumbersFor {
    public static void main(String[] args) {
        for(int i=0;i<=4;i++){
            System.out.println("Raj");
        }
    }
}
```

OUTPUT

```
Batman
Batman
Batman
Batman
Batman
```

EXPLANATION

- Here, the loop initializes `i` at 0 and continues while `i` is less than or equal to 4. This variant still results in 5 iterations (0, 1, 2, 3, 4), but it uses a zero-based index.



5. Display your name 5 times - 4

CODE

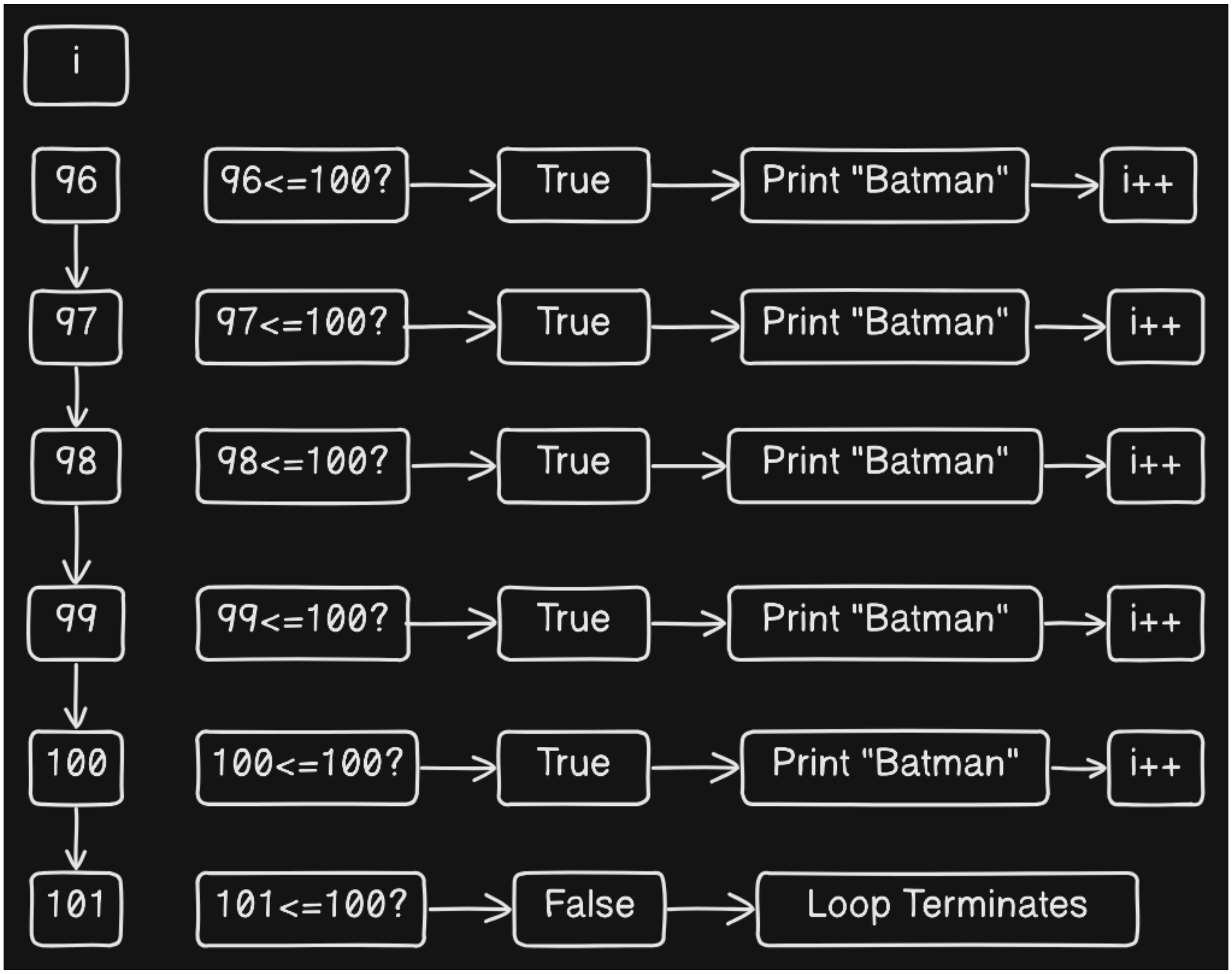
```
package loops;

public class NumbersFor {
    public static void main(String[] args) {
        for(int i=96;i<=100;i++){
            System.out.println("Batman");
        }
    }
}
```

OUTPUT

```
Batman
Batman
Batman
Batman
Batman
```

EXPLANATION

- In this loop, `i` starts at 96 and runs until it reaches 100. This still results in 5 iterations (96, 97, 98, 99, 100).
- The starting point for `i` doesn't really matter because we still get the same result. As long as the loop condition is set up correctly, we can choose any starting number for `i`. This means we can begin counting from 1, 0, or even 96, and we'll still end up printing the same number of times.
- If there is some kind of calculation involving `i` then we have to be careful with the initialization.



6. Where the value of `i` variable matters (In the first example it mattered too)

CODE

```
package loops;
```

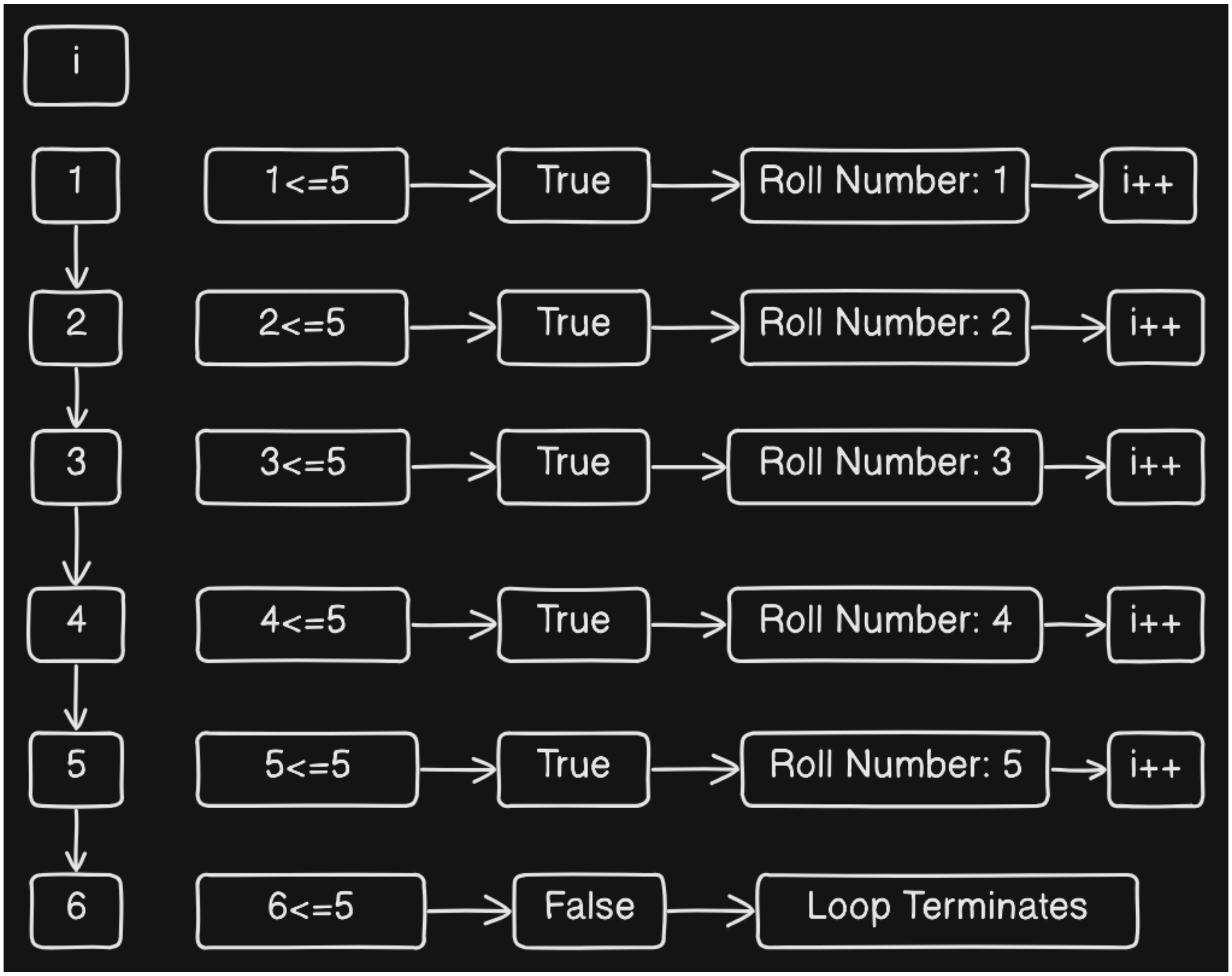
```
public class NumbersFor {
    public static void main(String[] args) {
        for(int i=1;i<=5;i++){
            System.out.println("Roll Number: "+i);
        }
    }
}
```

OUTPUT

```
Roll Number: 1
Roll Number: 2
Roll Number: 3
Roll Number: 4
Roll Number: 5
```

EXPLANATION

- Here we are printing the value of `i` , and if we initialize any other value of `i` , it will start printing from there. For ex: If we initialize `i` with 96, it will start printing Roll Number 96, 97, 98 and so on which we don't want.


i
1
1<=5
True
Roll Number: 1
i++
2
2<=5
True
Roll Number: 2
i++
3
3<=5
True
Roll Number: 3
i++
4
4<=5
True
Roll Number: 4
i++
5
5<=5
True
Roll Number: 5
i++
6
6<=5
False
Loop Terminates