
Do ... While Loop

A do...while is similar to a while loop in that it can be either a Count Controlled Loop or an Event Controlled Loop, but it is different in when it checks the conditional statement. A do...while is a post check, i.e. it checks the condition AFTER executing the code.

```
int counter = 0; //Have to initialize for Count Controlled Loop
do
{
    System.out.println(counter);
    counter++;
}while(counter < 25);
```

```
int choice;
```

```
do
{
    System.out.println("Enter 4 to leave the loop");
    choice = kb.nextInt();
}while(choice != 4);
```

Initial State

An event controlled loop doesn't need its conditional variable(s) to be initialized. The variable(s) simply has to be declared outside the do...while to handle variable scope.

Update State

After the loop's code has executed, the state must be updated. If the state isn't update, the loop may execute for forever!

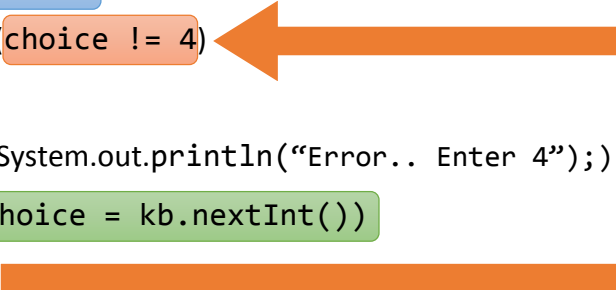
Conditional Statement

While the condition is true, the do .. while loop will continue to iterate. Only when the conditional statement evaluates to false will the do...while loop exit.

Warning

Make sure you are correctly setting the initial state. With a do...while() loop, the "initial state" is AFTER the code has executed. If you try to set the "initial state" outside the do..while() loop, that value won't be checked by the conditional statement, because it will be overwritten!

The do... while loop executes in the following steps:

1. Execute the code (`System.out.println("Error.. Enter 4");`)
2. Initial State (`choice = kb.nextInt()`)
3. Check the conditional statement (`choice != 4`)
 - a. True
 - i. Execute the code (`System.out.println("Error.. Enter 4");`)
 - ii. Update Counter (`choice = kb.nextInt()`)
 - iii. Go back to Step 2
 - b. False
 - i. Exit do... while loop

Do...While Loop Exercises

`DoWhileLoopExercises.java` & `DoWhileLoopAnswers.java`