While Loop

A while loop can be written as a Count Controlled Loop or an Event Controlled Loop, i.e. the number of times a while loop iterates can be set by a counter or by a conditional statement.

```
int counter = 0;//Declare/Initialize counter
while(counter < 25)
{
      System.out.println(counter);
      counter++;//Update Counter
}
```

```
System.out.println(“Please enter the number 4”);
int choice = kb.nextInt();//Initial State
while(choice != 4)
{
      System.out.println(“Error.. Enter 4”);
      choice = kb.nextInt();//Update State
}
```

Initial State

Before entering the while loop we must set the Initial State*. This way if the initial state is invalid, the program will not even enter the while loop.

Conditional Statement

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

Update State

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

Warning

Make sure you are correctly setting the initial state. If you are using user input, ask for user input BEFORE the while loop. Don’t force the while loop to execute! I.e. like (int choice = 0), you are forcing the while loop to enter and become a “post-check” loop like the do…while()

The while loop executes in the following steps:

1. Initial State (`int choice = kb.nextInt()`)
2. 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 while loop

While Loop Exercises

WhileLoopExercises.java & WhileLoopAnswers.java