

# The while Loop and Practice Problems

## CS 107

Stephen Majercik

### Use

To repeat execution of a statement or group of statements as long as a specified condition is satisfied. Note that the statement may not be executed even once if the condition is not satisfied when the `while` statement is reached.

### Form

```
while (boolean-expression)
    statement;
```

where `while` is a reserved word, `boolean-expression` is an expression that evaluates to `true` or `false`, and `statement` is a Java statement, or a group of statements enclosed by curly braces (a compound statement).

### Action

If the boolean expression is `true`, the specified statement, called the *body* of the loop, is executed. The boolean expression is then reevaluated and, if it is still `true`, the statement is executed again. This process of evaluating the boolean expression and executing the specified statement is repeated as long as the boolean expression is `true`. When it becomes `false`, repetition is terminated. Note that the statement must eventually force the specified condition to be *unsatisfied* so that the loop is terminated.

### Examples

#### Counter Controlled

```
System.out.print("How many scores do you have to input? ");
int numScores = r.readInt();
r.readLine();

int scoreCount = 1;
while (scoreCount <= numScores) {
    System.out.print("Enter score " + scoreCount + ": ");
    double score = r.readDouble();
    r.readLine();
    :
    :
    ++scoreCount; // same as: scoreCount = scoreCount + 1;
}
```

## Sentinel Controlled

```
System.out.println("Enter a negative score to signal the end of input.");
int scoreSum = 0;
int numScores = 0;
System.out.print("Score: ");
int score = r.readInt();
r.readLine();

while (score >= 0) {
    ++numScores; // same as: numScores = numScores + 1;
    scoreSum = scoreSum + score;
    System.out.print("Score: ");
    int score = r.readInt();
    r.readLine();
}
```

## Validating Input

```
int dimension = r.readInt();
r.readLine();

while ((dimension < MIN_DIMENSION) || (dimension > MAX_DIMENSION)) {
    System.out.print("Value of dimension must be from " + MIN_DIMENSION +
        " to " + MAX_DIMENSION + " inclusive. " + "
        " Please reenter: ");
    int dimension = r.readInt();
    r.readLine();
}
```

## Practice Problems

- What's wrong with the following `while` loop?

```
int counter = 0;
while {counter > 100}
    if (counter % 2 == 1)
        System.out.println(counter + " is odd.");
    else
        System.out.println(counter + " is odd.");
    ++counter; // same as: counter = counter + 1;
```

- Describe the output produced by these `while` loops:

```
a) int K = 5;
    int I = -2;
    while (I <= K) {
        I = I + 2;
        --K; // same as: K = K - 1;
        System.out.println(I + K);
    }
```

```
b) int number = 4;
    while (number >= 0)
        --number; // same as: number = number - 1;
        System.out.println(number);
```

- Write `while` loops to do the following:
  - Repeatedly print the value of the variable `xValue`, decreasing it by 0.5 each time, as long as `xValue` remains positive.
  - Print the square roots of the first 25 odd positive integers.
  - Repeats a block of code as long as the user indicates they want it to.
  - Drive the user crazy by insisting they re-enter a particular input no matter what they enter. Be creative...