While and Do-While; Switch

Mitsu Ogihara

Department of Computer Science
University of Miami

Table of Contents

While Loops

The while loop is a structure for repeating a block of statements as long as a certain continuation condition is met

The structure for a while loop is simple:

```
while (CONDITION) {
   .  .  .
}
```

A while-loop can be simulated by a for loop, vice versa

```
while ( X ) { W; }
```

is equivalent to:

```
for ( ; X ; ) { W; }
```

A while-loop can be simulated by a for loop, vice versa

```
for ( A ; X ; B ) { W; }
```

is equivalent to:

```
A;
while ( X ) { W; B; }
```

Simple Example

Set the variable `value` to 10 and then decrease the value by one repeatedly, until it becomes 0

Simple Example

Set the variable `value` to 10 and then decrease the value by one repeatedly, until it becomes 0

```
1  public class CountDownWhile {
2    public static void main( String[] args ) {
3      int value = 10;
4      while ( value > 0 ) {
5        System.out.println( value );
6        value --;
7      }
8    }
9  }
```

The while loop; the condition is `value>0` and the body contains `value--`

Example 1: Collatz Conjecture

The Collatz Conjecture states that any positive integer can be converted to 1 by repeating the following process:

Example 1: Collatz Conjecture

The Collatz Conjecture states that any positive integer can be converted to 1 by repeating the following process:

- If the number is an even number divide it by 2.
- If the number is an odd number, multiply it by 3 and then add 1.

Example 1: Collatz Conjecture

The Collatz Conjecture states that any positive integer can be converted to 1 by repeating the following process:

- If the number is an even number divide it by 2.
- If the number is an odd number, multiply it by 3 and then add 1.

For example, 7 can be turned into 1 by:

$$7 \to 22 \to 11 \to 34 \to 17 \to 52 \to 26 \to 13 \to$$
$$40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$$

Writing a Code for Testing the Conjecture for Numbers Entered

- Create a method that tests the conjecture on a given number by generating the chain of numbers generated from the input
  - Use a method `update` for generating the next element, given the current element; e.g., $2 \rightarrow 1$ and $3 \rightarrow 10$
  - Use the current element, the previous element, the current position in the chain
  - Use a while loop to generate to update the three various until the current element becomes 1
- Receive an int from the user and execute the test
- The user can perform tests on as many input numbers as he/she wishes; entering a number less than 1 quits the program

Collatz Conjecture

```
1  import java.util.Scanner;
2  public class Collatz {
3    //--- generate the next element
4    public static int update( int number ) {
5      if ( number % 2 == 0 ) {
6        return number / 2;
7      }
8      return number * 3 + 1;
9    }
```

The method for performing one step of modification conversion

Collatz Conjecture

```
1  import java.util.Scanner;
2  public class Collatz {
3    //--- generate the next element
4    public static int update( int number ) {
5      if ( number % 2 == 0 ) {
6        return number / 2;
7      }
8      return number * 3 + 1;
9    }
```

If `number` is even, return the number divided by 2

Collatz Conjecture

```
1  import java.util.Scanner;
2  public class Collatz {
3    //--- generate the next element
4    public static int update( int number ) {
5      if ( number % 2 == 0 ) {
6        return number / 2;
7      }
8      return number * 3 + 1;
9    }
```

If `number` is odd, return the number times 3 + 1; Note the lack of `else`

Collatz Conjecture

```
10    //--- generate the chain to perform the test
11    public static void test( int number ) {
12      System.out.println( "input is " + number );
13      int current = number, previous, round = 0;
14      while ( current > 1 ) {
15        round ++;
16        previous = current;
17        current = update( previous );
18        System.out.printf( "%4d: %7d -> %d%n",
19            round, previous, current );
20      }
21    }
```

The method for testing the hypothesis for one number
It receives an integer `number` as a parameter

Collatz Conjecture

```
10    //--- generate the chain to perform the test
11    public static void test( int number ) {
12      System.out.println( "input is " + number );
13      int current = number, previous, round = 0;
14      while ( current > 1 ) {
15        round ++;
16        previous = current;
17        current = update( previous );
18        System.out.printf( "%4d: %7d -> %d%n",
19            round, previous, current );
20      }
21    }
```

Print the input number on the screen

Collatz Conjecture

```
10    //--- generate the chain to perform the test
11    public static void test( int number ) {
12      System.out.println( "input is " + number );
13      int current = number, previous, round = 0;
14      while ( current > 1 ) {
15        round ++;
16        previous = current;
17        current = update( previous );
18        System.out.printf( "%4d: %7d -> %d%n",
19            round, previous, current );
20      }
21    }
```

The three variables to use

Collatz Conjecture

```
10    //--- generate the chain to perform the test
11    public static void test( int number ) {
12      System.out.println( "input is " + number );
13      int current = number, previous, round = 0;
14      while ( current > 1 ) {
15        round ++;
16        previous = current;
17        current = update( previous );
18        System.out.printf( "%4d: %7d -> %d%n",
19            round, previous, current );
20      }
21    }
```

While the number is greater than 1, perform the inside the loop

Collatz Conjecture

```
10    //--- generate the chain to perform the test
11    public static void test( int number ) {
12      System.out.println( "input is " + number );
13      int current = number, previous, round = 0;
14      while ( current > 1 ) {
15        round ++;
16        previous = current;
17        current = update( previous );
18        System.out.printf( "%4d: %7d -> %d%n",
19            round, previous, current );
20      }
21    }
```

Increase the round by 1

Collatz Conjecture

```
10    //--- generate the chain to perform the test
11    public static void test( int number ) {
12      System.out.println( "input is " + number );
13      int current = number, previous, round = 0;
14      while ( current > 1 ) {
15        round ++;
16        previous = current;
17        current = update( previous );
18        System.out.printf( "%4d: %7d -> %d%n",
19            round, previous, current );
20      }
21    }
```

Store the value of `current` into `previous`

Collatz Conjecture

```
10  //--- generate the chain to perform the test
11  public static void test( int number ) {
12    System.out.println( "input is " + number );
13    int current = number, previous, round = 0;
14    while ( current > 1 ) {
15      round ++;
16      previous = current;
17      current = update( previous );
18      System.out.printf( "%4d: %7d -> %d%n",
19          round, previous, current );
20    }
21  }
```

Update `current` with the value of the `update` method with `previous` as the parameter value

Collatz Conjecture

```
10    //--- generate the chain to perform the test
11    public static void test( int number ) {
12      System.out.println( "input is " + number );
13      int current = number, previous, round = 0;
14      while ( current > 1 ) {
15        round ++;
16        previous = current;
17        current = update( previous );
18        System.out.printf( "%4d: %7d -> %d%n",
19            round, previous, current );
20      }
21    }
```

Inside the loop, print the round , the previous element, and the current element

Collatz Conjecture

```
22    //--- main
23    public static void main( String[] args ) {
24      System.out.println( "---Testing Collatz Conjecture---" );
25      Scanner console = new Scanner( System.in );
26      int input = 1;
27      while ( input > 0 ) {
28        System.out.print( "Enter input > 1 : " );
29        input = console.nextInt();
30        if ( input > 1 ) {
31          test( input );
32        }
33      }
34    }
35  }
```

Print a message

Collatz Conjecture

```
22    //--- main
23    public static void main( String[] args ) {
24      System.out.println( "---Testing Collatz Conjecture---" );
25      Scanner console = new Scanner( System.in );
26      int input = 1;
27      while ( input > 0 ) {
28        System.out.print( "Enter input > 1 : " );
29        input = console.nextInt();
30        if ( input > 1 ) {
31          test( input );
32        }
33      }
34    }
35  }
```

Create a scanner

Collatz Conjecture

```
22    //--- main
23    public static void main( String[] args ) {
24      System.out.println( "---Testing Collatz Conjecture---" );
25      Scanner console = new Scanner( System.in );
26      int input = 1;
27      while ( input > 0 ) {
28        System.out.print( "Enter input > 1 : " );
29        input = console.nextInt();
30        if ( input > 1 ) {
31          test( input );
32        }
33      }
34    }
35  }
```

Use the variable `input` for the input; the initial value of the variable is 1 and so the body will be executed at least once

Collatz Conjecture

```
22    //--- main
23    public static void main( String[] args ) {
24      System.out.println( "---Testing Collatz Conjecture---" );
25      Scanner console = new Scanner( System.in );
26      int input = 1;
27      while ( input > 0 ) {
28        System.out.print( "Enter input > 1 : " );
29        input = console.nextInt();
30        if ( input > 1 ) {
31          test( input );
32        }
33      }
34    }
35  }
```

Receive input from the user and if the input is greater than 1 execute the test; otherwise, the loop quits

Binary Numbers and Decimal Numbers

Suppose we want to convert a given positive decimal number to a binary number

Let v be the input number
We can use the following idea:

- Set a String s to the empty string
- Set an integer c to v
- While c is not 0 do the following:
  - Insert at the beginning of s a 0 if c is even and a 1 otherwise
  - Set c to $c/2$

ConvertToBinary

```
1  import java.util.*;
2  public class ConvertToBinaryInAction {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      int value = 0;
6      while ( value >=0 ) {
7        System.out.print( "Enter a nonnegative integer: " );
8        value = console .nextInt();
9        if ( value >= 0 ) {
10         System.out.printf( "%d --> %s%n",
11             value, convert( value ) );
12       }
13     }
14   }
```

Create a keyboard `Scanner`

ConvertToBinary

```
1  import java.util.*;
2  public class ConvertToBinaryInAction {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      int value = 0;
6      while ( value >=0 ) {
7        System.out.print( "Enter a nonnegative integer: " );
8        value = console .nextInt();
9        if ( value >= 0 ) {
10         System.out.printf( "%d --> %s%n",
11             value, convert( value ) );
12       }
13     }
14   }
```

Initialize the variable `value` with 0 to get started in the loop

ConvertToBinary

```
1  import java.util.*;
2  public class ConvertToBinaryInAction {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      int value = 0;
6      while ( value >=0 ) {
7        System.out.print( "Enter a nonnegative integer: " );
8        value = console .nextInt();
9        if ( value >= 0 ) {
10         System.out.printf( "%d --> %s%n",
11             value, convert( value ) );
12       }
13     }
14   }
```

As long as `value` is nonnegative the loop goes on

ConvertToBinary

```
1  import java.util.*;
2  public class ConvertToBinaryInAction {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      int value = 0;
6      while ( value >=0 ) {
7        System.out.print( "Enter a nonnegative integer: " );
8        value = console .nextInt();
9        if ( value >= 0 ) {
10         System.out.printf( "%d --> %s%n",
11             value, convert( value ) );
12       }
13     }
14   }
```

Prompt the user to receive the `value`

ConvertToBinary

```
1  import java.util.*;
2  public class ConvertToBinaryInAction {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      int value = 0;
6      while ( value >=0 ) {
7        System.out.print( "Enter a nonnegative integer: " );
8        value = console .nextInt();
9        if ( value >= 0 ) {
10         System.out.printf( "%d --> %s%n",
11             value, convert( value ) );
12       }
13     }
14   }
```

If the value is nonnegative, print the value and the result of conversion
The value is printed as a while number (`"%d"`) and the result as a `String` (`"%s"`)

ConvertToBinary

```
15    public static String convert( int value ) {
16      if ( value == 0 ) {
17        return "0";
18      }
19      String binary = "", bit = "";
20      System.out.printf( "%10s, %s%n", "Digits", "Binary" );
21      while ( value > 0 ) {
22        if ( value % 2 == 1 ) {
23          binary = "1" + binary;
24        }
25        else {
26          binary = "0" + binary;
27        }
28        value /= 2;
29        System.out.printf( "%10d, %s%n", value, binary );
30      }
31      return binary;
32    }
33  }
```

The conversion method receives as a parameter an `int value`

ConvertToBinary

```
15    public static String convert( int value ) {
16      if ( value == 0 ) {
17        return "0";
18      }
19      String binary = "", bit = "";
20      System.out.printf( "%10s, %s%n", "Digits", "Binary" );
21      while ( value > 0 ) {
22        if ( value % 2 == 1 ) {
23          binary = "1" + binary;
24        }
25        else {
26          binary = "0" + binary;
27        }
28        value /= 2;
29        System.out.printf( "%10d, %s%n", value, binary );
30      }
31      return binary;
32    }
33  }
```

If `value` is equal to 0, return `"0"`

ConvertToBinary

```
15    public static String convert( int value ) {
16      if ( value == 0 ) {
17        return "0";
18      }
19      String binary = "", bit = "";
20      System.out.printf( "%10s, %s%n", "Digits", "Binary" );
21      while ( value > 0 ) {
22        if ( value % 2 == 1 ) {
23          binary = "1" + binary;
24        }
25        else {
26          binary = "0" + binary;
27        }
28        value /= 2;
29        System.out.printf( "%10d, %s%n", value, binary );
30      }
31      return binary;
32    }
33  }
```

Otherwise, build in `binary` the return `String`; use bit `bit` to represent the bit computed

ConvertToBinary

```
15    public static String convert( int value ) {
16      if ( value == 0 ) {
17        return "0";
18      }
19      String binary = "", bit = "";
20      System.out.printf( "%10s, %s%n", "Digits", "Binary" );
21      while ( value > 0 ) {
22        if ( value % 2 == 1 ) {
23          binary = "1" + binary;
24        }
25        else {
26          binary = "0" + binary;
27        }
28        value /= 2;
29        System.out.printf( "%10d, %s%n", value, binary );
30      }
31      return binary;
32    }
33  }
```

Print the header

ConvertToBinary

```
15    public static String convert( int value ) {
16      if ( value == 0 ) {
17        return "0";
18      }
19      String binary = "", bit = "";
20      System.out.printf( "%10s, %s%n", "Digits", "Binary" );
21      while ( value > 0 ) {
22        if ( value % 2 == 1 ) {
23          binary = "1" + binary;
24        }
25        else {
26          binary = "0" + binary;
27        }
28        value /= 2;
29        System.out.printf( "%10d, %s%n", value, binary );
30      }
31      return binary;
32    }
33  }
```

As long as the value is greater than 0, do the following

ConvertToBinary

```
15    public static String convert( int value ) {
16      if ( value == 0 ) {
17        return "0";
18      }
19      String binary = "", bit = "";
20      System.out.printf( "%10s, %s%n", "Digits", "Binary" );
21      while ( value > 0 ) {
22        if ( value % 2 == 1 ) {
23          binary = "1" + binary;
24        }
25        else {
26          binary = "0" + binary;
27        }
28        value /= 2;
29        System.out.printf( "%10d, %s%n", value, binary );
30      }
31      return binary;
32    }
33  }
```

Insert before `binary`: `'1'` if `value` is odd and `'0'` otherwise

ConvertToBinary

```
15    public static String convert( int value ) {
16      if ( value == 0 ) {
17        return "0";
18      }
19      String binary = "", bit = "";
20      System.out.printf( "%10s, %s%n", "Digits", "Binary" );
21      while ( value > 0 ) {
22        if ( value % 2 == 1 ) {
23          binary = "1" + binary;
24        }
25        else {
26          binary = "0" + binary;
27        }
28        value /= 2;
29        System.out.printf( "%10d, %s%n", value, binary );
30      }
31      return binary;
32    }
33  }
```

Divide value by 2; output the current status

ConvertToBinary

```
15    public static String convert( int value ) {
16      if ( value == 0 ) {
17        return "0";
18      }
19      String binary = "", bit = "";
20      System.out.printf( "%10s, %s%n", "Digits", "Binary" );
21      while ( value > 0 ) {
22        if ( value % 2 == 1 ) {
23          binary = "1" + binary;
24        }
25        else {
26          binary = "0" + binary;
27        }
28        value /= 2;
29        System.out.printf( "%10d, %s%n", value, binary );
30      }
31      return binary;
32    }
33  }
```

After quitting the loop return the string

Table of Contents

Do-while Loops

The do-while loop is a variant of while loop in which the execution of the loop body is forced at the beginning

The structure for a while loop is simple:

```
do {
   .  .  .
} while (CONDITION);
```

A do-while loop can be simulated by a while loop

```
do { W; } while ( X );
```

is equivalent to:

```
W;
while ( X ) { W; }
```

Do-while Loops

The do-while loop is a variant of while loop in which the execution of the loop body is forced at the beginning

The structure for a while loop is simple:

```
do {
   .  .  .
} while (CONDITION);
```

Example: Receive Numbers and Sum Them Up

Here is a simple application with which to receive numbers from the user and calculate the running total of the numbers

When 0 is entered the calculation terminates

Example 1: Receive and Add Numbers

```
1  import java.util.*;
2  public class ReceiveAndAddAlt {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      // initialize the total to 0
6      int sum = 0;
7      // this is the number to receive, set to -1 initially
8      int number = 0;
9      // repeat the loop body until the entered number is 0
10     do {
11       System.out.printf( "The current total is %d%n"
12           + "Enter a number to add (0 to quit): ", sum );
13       number = console.nextInt();
14       // add the number to the sum
15       sum += number;
16     } while ( number != 0 );
17     System.out.println( "The total is " + sum );
18   }
19 }
```

The total and the number to receive

Example 1: Receive and Add Numbers

```
1  import java.util.*;
2  public class ReceiveAndAddAlt {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      // initialize the total to 0
6      int sum = 0;
7      // this is the number to receive, set to -1 initially
8      int number = 0;
9      // repeat the loop body until the entered number is 0
10     do {
11       System.out.printf( "The current total is %d%n"
12           + "Enter a number to add (0 to quit): ", sum );
13       number = console.nextInt();
14       // add the number to the sum
15       sum += number;
16     } while ( number != 0 );
17     System.out.println( "The total is " + sum );
18   }
19 }
```

The do-while loop

Example 1: Receive and Add Numbers

```
1  import java.util.*;
2  public class ReceiveAndAddAlt {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      // initialize the total to 0
6      int sum = 0;
7      // this is the number to receive, set to -1 initially
8      int number = 0;
9      // repeat the loop body until the entered number is 0
10     do {
11       System.out.printf( "The current total is %d%n"
12           + "Enter a number to add (0 to quit): ", sum );
13       number = console.nextInt();
14       // add the number to the sum
15       sum += number;
16     } while ( number != 0 );
17     System.out.println( "The total is " + sum );
18   }
19 }
```

The number is received and add it to the sum
Note that 0 is the number of trigger termination but also added to the sum

Example 2: Receive and Add Numbers, Using a While Loop

```
1  import java.util.*;
2  public class ReceiveAndAdd {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      // initialize the total to 0
6      int sum = 0;
7      // this is the number to receive, set to -1 initially
8      int number = - 1;
9      // repeat the loop body until the entered number is 0
10     while ( number != 0 ) {
11       System.out.printf( "The current total is %d%n"
12           + "Enter a number to add (0 to quit): ", sum );
13       number = console.nextInt();
14       sum += number;
15     }
16     System.out.println( "The total is " + sum );
17   }
18 }
```

The total, initialized to 0 as before

Example 2: Receive and Add Numbers, Using a While Loop

```
1  import java.util.*;
2  public class ReceiveAndAdd {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      // initialize the total to 0
6      int sum = 0;
7      // this is the number to receive, set to -1 initially
8      int number = - 1;
9      // repeat the loop body until the entered number is 0
10     while ( number != 0 ) {
11       System.out.printf( "The current total is %d%n"
12           + "Enter a number to add (0 to quit): ", sum );
13       number = console.nextInt();
14       sum += number;
15     }
16     System.out.println( "The total is " + sum );
17   }
18 }
```

The input, initialized to any number **other than 0**
Here -1 is chosen

Example 2: Receive and Add Numbers, Using a While Loop

```
1  import java.util.*;
2  public class ReceiveAndAdd {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      // initialize the total to 0
6      int sum = 0;
7      // this is the number to receive, set to -1 initially
8      int number = - 1;
9      // repeat the loop body until the entered number is 0
10     while ( number != 0 ) {
11       System.out.printf( "The current total is %d%n"
12           + "Enter a number to add (0 to quit): ", sum );
13       number = console.nextInt();
14       sum += number;
15     }
16     System.out.println( "The total is " + sum );
17   }
18 }
```

The while loop

Example 2: Receive and Add Numbers, Using a While Loop

```
1  import java.util.*;
2  public class ReceiveAndAdd {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      // initialize the total to 0
6      int sum = 0;
7      // this is the number to receive, set to -1 initially
8      int number = - 1;
9      // repeat the loop body until the entered number is 0
10     while ( number != 0 ) {
11       System.out.printf( "The current total is %d%n"
12           + "Enter a number to add (0 to quit): ", sum );
13       number = console.nextInt();
14       sum += number;
15     }
16     System.out.println( "The total is " + sum );
17   }
18 }
```

The number is received and add it to the sum

Table of Contents

Random

Class **Random** is a class that provide methods for generating random numbers.

You need to declare: `import java.util.Random;` or `import java.util.*;`

To use you need to create a Random type, e.g., by:

```
Random r = new Random();
```

After creation you can execute methods on the Random type

Methods in Random

- `nextInt()` : returns a random int value;
- `nextInt(int n)` : returns a random int between 0 and $n - 1$
- `nextDouble()` : returns a random real between 0 and 1 (0 included, 1 excluded)

Example: Throw Dice

Throw a die a number of times specified by the user and count how many times each face shows

The code

```
1  import java.util.*;
2  public class ThrowDice {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      Random rand = new Random();
6      int numberOfRounds, die;
7      int count1 = 0, count2 = 0, count3 = 0;
8      int count4 = 0, count5 = 0, count6 = 0;
9      System.out.print( "How many throws ? " );
10     numberOfRounds = console.nextInt();
```

Declaring a Random type

The code

```
1  import java.util.*;
2  public class ThrowDice {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      Random rand = new Random();
6      int numberOfRounds, die;
7      int count1 = 0, count2 = 0, count3 = 0;
8      int count4 = 0, count5 = 0, count6 = 0;
9      System.out.print( "How many throws ? " );
10     numberOfRounds = console.nextInt();
```

Counters

The code

```
1  import java.util.*;
2  public class ThrowDice {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      Random rand = new Random();
6      int numberOfRounds, die;
7      int count1 = 0, count2 = 0, count3 = 0;
8      int count4 = 0, count5 = 0, count6 = 0;
9      System.out.print( "How many throws ? " );
10     numberOfRounds = console.nextInt();
```

Receive input from user

The code

```
11      for ( int round = 1; round <= numberOfRounds; round ++ ) {
12        die = rand.nextInt( 6 ) + 1;
13        if ( die == 1 ) {
14          count1 ++;
15        }
16        else if ( die == 2 ) {
17          count2 ++;
18        }
19        else if ( die == 3 ) {
20          count3 ++;
21        }
22        else if ( die == 4 ) {
23          count4 ++;
24        }
25        else if ( die == 5 ) {
26          count5 ++;
27        }
28        else {
29          count6 ++;
30        }
31      }
```

for loop

The code

```
11      for ( int round = 1; round <= numberOfRounds; round ++ ) {
12        die = rand.nextInt( 6 ) + 1;
13        if ( die == 1 ) {
14          count1 ++;
15        }
16        else if ( die == 2 ) {
17          count2 ++;
18        }
19        else if ( die == 3 ) {
20          count3 ++;
21        }
22        else if ( die == 4 ) {
23          count4 ++;
24        }
25        else if ( die == 5 ) {
26          count5 ++;
27        }
28        else {
29          count6 ++;
30        }
31      }
```

Processing the count

The code

```
32      System.out.println( count1 );
33      System.out.println( count2 );
34      System.out.println( count3 );
35      System.out.println( count4 );
36      System.out.println( count5 );
37      System.out.println( count6 );
38    }
39  }
```

Output generation

Table of Contents

An Infinite While Loops

An infinite while loop can be designed by setting the condition to `true`

The structure for an infinite while-loop is:

```
while (true) {
   .   .   .
}
```

An Infinite While Loops

An infinite while loop can be designed by setting the condition to `true`

The structure for an infinite while-loop is:

```
while (true) {
   .   .   .
}
```

This is not meant to be a true infinite loop and so there should be a mechanism inside the body for exiting the loop

An Infinite While Loops

An infinite while loop can be designed by setting the condition to `true`

The structure for an infinite while-loop is:

```
while (true) {
   .   .   .
}
```

This is not meant to be a true infinite loop and so there should be a mechanism inside the body for exiting the loop It is also possible to write an

infinite do-while loop using `true` as the condition

Quitting an Infinite Loop

There are two major ways to quit an infinite loop

Quitting an Infinite Loop

There are two major ways to quit an infinite loop

- `return` : the method containing the infinite while loop will be terminated right there with a return value if necessary

Quitting an Infinite Loop

There are two major ways to quit an infinite loop

- `return` : the method containing the infinite while loop will be terminated right there with a return value if necessary
- `break` : the innermost loop containing the break statement will be terminated

Substituting while (true)

It is of course possible to convert the `while (true)` to a while loop with a condition that may not be true all the time
Consider the following code:

```
1  while (true) {
2    BLOCK_1;
3    if (CONDITION_1) {
4      BLOCK_2;
5      if (CONDITION_2) {
6        break;
7      }
8      BLOCK_3;
9    }
10   BLOCK_4;
11 }
```

Without Infinite Looping

The code can be converted one with a loop condition:

```
1  boolean stillOn = true;
2  while (stillOn) {
3    BLOCK_1;
4    if (CONDITION_1) {
5      BLOCK_2;
6      if (CONDITION_2) {
7        stillOn = false;
8      }
9      if (stillOn) {
10       BLOCK_3;
11     }
12   }
13   if (stillOn) {
14     BLOCK_4;
15   }
16 }
```

Table of Contents

What Is the Switch Statement?

The **switch** statement is a mechanism for controlling the flow of program based on the value of one variable, where the variable is either a whole number type, including char, or a String, by organizing the branches in a more understandable way

The Syntax

Let `A1`, ..., `Am` and and `B1`, ..., `Bn` be some $m + n$ distinct literals that variable `x` might take upon, where both `m` and `n` can be equal to 0
Using these values a switch statement on `x` takes the following form:

```
switch (x) {
case A1:  STATEMENTS-A1
.  .  .
case Am:  STATEMENTS-Am
default:  STATEMENTS-default
case B1:  STATEMENTS-B1
.  .  .
case Bn:  STATEMENTS-Bn
}
```

Here `STATEMENTS` are any number (including 0) of statements

Anatomy of the Switch-statement

The `case` and `default` are “anchors” or “labels” embedded in the code
Without them, the code should be like:

```
STATEMENTS-A1
.  .  .
STATEMENTS-Am
STATEMENTS-default
STATEMENTS-B1
.  .  .
STATEMENTS-Bn
}
```

The “switch” allows one jump into the code depending on the value of `x`

Jumping into the Code

At the start of “switch” if the `x` is equal to one of `A1`, . . . , `Am` and and `B1`, . . . , `Bn`, then the execution starts from the code location corresponding to the label; otherwise, it starts from the location indicated by “default”

```
switch (x) {
case A1:  STATEMENTS-A1
.  .  .
case Am:  STATEMENTS-Am
default:  STATEMENTS-default
case B1:  STATEMENTS-B1
.  .  .
case Bn:  STATEMENTS-Bn
}
```

Jumping out from Switch

During the execution of “switch” statement, if `break;` is encountered, then the remainder of the “switch” block will be ignored

Example 1 SelectionByChar

```
1  import java.util.*;
2  public class SelectionByChar {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  A(ge), G(ender), N(ame), P(hone): " );
14     String response = console.next();
15     switch ( response.charAt( 0 ) ) {
16     case 'A': System.out.println( "The age is " + age ); break;
17     case 'G': System.out.println( "The gender is " + gender ); break;
18     case 'N': System.out.println( "The name is " + name ); break;
19     case 'P': System.out.println( "The phone " + phone ); break;
20     default:  System.out.println( "Unsupported selection!" );
21     }
22   }
23 }
```

Create a keyboard `Scanner`

Example 1 `SelectionByChar`

```
1  import java.util.*;
2  public class SelectionByChar {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  A(ge), G(ender), N(ame), P(hone): " );
14     String response = console.next();
15     switch ( response.charAt( 0 ) ) {
16     case 'A': System.out.println( "The age is " + age ); break;
17     case 'G': System.out.println( "The gender is " + gender ); break;
18     case 'N': System.out.println( "The name is " + name ); break;
19     case 'P': System.out.println( "The phone " + phone ); break;
20     default:  System.out.println( "Unsupported selection!" );
21     }
22   }
23 }
```

Receive name as a `String` object

Example 1 SelectionByChar

```
1  import java.util.*;
2  public class SelectionByChar {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  A(ge), G(ender), N(ame), P(hone): " );
14     String response = console.next();
15     switch ( response.charAt( 0 ) ) {
16     case 'A': System.out.println( "The age is " + age ); break;
17     case 'G': System.out.println( "The gender is " + gender ); break;
18     case 'N': System.out.println( "The name is " + name ); break;
19     case 'P': System.out.println( "The phone " + phone ); break;
20     default:  System.out.println( "Unsupported selection!" );
21     }
22   }
23 }
```

Receive age as an `int`

Example 1 `SelectionByChar`

```
1  import java.util.*;
2  public class SelectionByChar {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  A(ge), G(ender), N(ame), P(hone): " );
14     String response = console.next();
15     switch ( response.charAt( 0 ) ) {
16     case 'A': System.out.println( "The age is " + age ); break;
17     case 'G': System.out.println( "The gender is " + gender ); break;
18     case 'N': System.out.println( "The name is " + name ); break;
19     case 'P': System.out.println( "The phone " + phone ); break;
20     default:  System.out.println( "Unsupported selection!" );
21     }
22   }
23 }
```

Receive gender as a `String` object

Example 1 SelectionByChar

```
1  import java.util.*;
2  public class SelectionByChar {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  A(ge), G(ender), N(ame), P(hone): " );
14     String response = console.next();
15     switch ( response.charAt( 0 ) ) {
16     case 'A': System.out.println( "The age is " + age ); break;
17     case 'G': System.out.println( "The gender is " + gender ); break;
18     case 'N': System.out.println( "The name is " + name ); break;
19     case 'P': System.out.println( "The phone " + phone ); break;
20     default:  System.out.println( "Unsupported selection!" );
21     }
22   }
23 }
```

Receive phone as a `String` object

Example 1 SelectionByChar

```
1  import java.util.*;
2  public class SelectionByChar {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  A(ge), G(ender), N(ame), P(hone): " );
14     String response = console.next();
15     switch ( response.charAt( 0 ) ) {
16     case 'A': System.out.println( "The age is " + age ); break;
17     case 'G': System.out.println( "The gender is " + gender ); break;
18     case 'N': System.out.println( "The name is " + name ); break;
19     case 'P': System.out.println( "The phone " + phone ); break;
20     default:  System.out.println( "Unsupported selection!" );
21     }
22   }
23 }
```

Receive input from user

Example 1 SelectionByChar

```
1  import java.util.*;
2  public class SelectionByChar {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  A(ge), G(ender), N(ame), P(hone): " );
14     String response = console.next();
15     switch ( response.charAt( 0 ) ) {
16     case 'A': System.out.println( "The age is " + age ); break;
17     case 'G': System.out.println( "The gender is " + gender ); break;
18     case 'N': System.out.println( "The name is " + name ); break;
19     case 'P': System.out.println( "The phone " + phone ); break;
20     default:  System.out.println( "Unsupported selection!" );
21     }
22   }
23 }
```

Select action on the first letter of the input

Example 1 `SelectionByChar`

```
1  import java.util.*;
2  public class SelectionByChar {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  A(ge), G(ender), N(ame), P(hone): " );
14     String response = console.next();
15     switch ( response.charAt( 0 ) ) {
16     case 'A': System.out.println( "The age is " + age ); break;
17     case 'G': System.out.println( "The gender is " + gender ); break;
18     case 'N': System.out.println( "The name is " + name ); break;
19     case 'P': System.out.println( "The phone " + phone ); break;
20     default:  System.out.println( "Unsupported selection!" );
21     }
22   }
23 }
```

The literal after case is a `char` literal

Example 2: SelectionByString

```
1  import java.util.*;
2  public class SelectionByString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  Age, Gender, Name, Phone: " );
14     String response = console.next();
15     switch ( response ) {
16     case "Age": System.out.println( "The age is " + age );
17         break;
18     case "Gender": System.out.println( "The gender is " + gender );
19         break;
20     case "Name": System.out.println( "The name is " + name );
21         break;
22     case "Phone": System.out.println( "The phone " + phone );
23         break;
24     default: System.out.println( "Unsupported selection!" );
25     }
26   }
27 }
```

Create a keyboard `Scanner`

Example 2: SelectionByString

```
1  import java.util.*;
2  public class SelectionByString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  Age, Gender, Name, Phone: " );
14     String response = console.next();
15     switch ( response ) {
16     case "Age": System.out.println( "The age is " + age );
17         break;
18     case "Gender": System.out.println( "The gender is " + gender );
19         break;
20     case "Name": System.out.println( "The name is " + name );
21         break;
22     case "Phone": System.out.println( "The phone " + phone );
23         break;
24     default: System.out.println( "Unsupported selection!" );
25     }
26   }
27 }
```

Receive name as a `String` object

Example 2: `SelectionByString`

```
1  import java.util.*;
2  public class SelectionByString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  Age, Gender, Name, Phone: " );
14     String response = console.next();
15     switch ( response ) {
16     case "Age": System.out.println( "The age is " + age );
17         break;
18     case "Gender": System.out.println( "The gender is " + gender );
19         break;
20     case "Name": System.out.println( "The name is " + name );
21         break;
22     case "Phone": System.out.println( "The phone " + phone );
23         break;
24     default: System.out.println( "Unsupported selection!" );
25     }
26   }
27 }
```

Receive age as an `int`

Example 2: `SelectionByString`

```
1  import java.util.*;
2  public class SelectionByString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  Age, Gender, Name, Phone: " );
14     String response = console.next();
15     switch ( response ) {
16     case "Age": System.out.println( "The age is " + age );
17         break;
18     case "Gender": System.out.println( "The gender is " + gender );
19         break;
20     case "Name": System.out.println( "The name is " + name );
21         break;
22     case "Phone": System.out.println( "The phone " + phone );
23         break;
24     default: System.out.println( "Unsupported selection!" );
25     }
26   }
27 }
```

Receive gender as a `String` object

Example 2: SelectionByString

```
1  import java.util.*;
2  public class SelectionByString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  Age, Gender, Name, Phone: " );
14     String response = console.next();
15     switch ( response ) {
16     case "Age": System.out.println( "The age is " + age );
17         break;
18     case "Gender": System.out.println( "The gender is " + gender );
19         break;
20     case "Name": System.out.println( "The name is " + name );
21         break;
22     case "Phone": System.out.println( "The phone " + phone );
23         break;
24     default: System.out.println( "Unsupported selection!" );
25     }
26   }
27 }
```

Receive phone as a `String` object

Example 2: SelectionByString

```
1  import java.util.*;
2  public class SelectionByString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  Age, Gender, Name, Phone: " );
14     String response = console.next();
15     switch ( response ) {
16     case "Age": System.out.println( "The age is " + age );
17         break;
18     case "Gender": System.out.println( "The gender is " + gender );
19         break;
20     case "Name": System.out.println( "The name is " + name );
21         break;
22     case "Phone": System.out.println( "The phone " + phone );
23         break;
24     default: System.out.println( "Unsupported selection!" );
25     }
26   }
27 }
```

Receive input from user

Example 2: SelectionByString

```
1  import java.util.*;
2  public class SelectionByString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  Age, Gender, Name, Phone: " );
14     String response = console.next();
15     switch ( response ) {
16     case "Age": System.out.println( "The age is " + age );
17         break;
18     case "Gender": System.out.println( "The gender is " + gender );
19         break;
20     case "Name": System.out.println( "The name is " + name );
21         break;
22     case "Phone": System.out.println( "The phone " + phone );
23         break;
24     default: System.out.println( "Unsupported selection!" );
25     }
26   }
27 }
```

Select action on the first letter of the input

Example 2: SelectionByString

```
1  import java.util.*;
2  public class SelectionByString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Type name: " );
6      String name = console.next();
7      System.out.print( "Type age: " );
8      int age = console.nextInt();
9      System.out.print( "Type gender: " );
10     String gender = console.next();
11     System.out.print( "Type phone (use dash): " );
12     String phone = console.next();
13     System.out.print( "  Age, Gender, Name, Phone: " );
14     String response = console.next();
15     switch ( response ) {
16     case "Age": System.out.println( "The age is " + age );
17         break;
18     case "Gender": System.out.println( "The gender is " + gender );
19         break;
20     case "Name": System.out.println( "The name is " + name );
21         break;
22     case "Phone": System.out.println( "The phone " + phone );
23         break;
24     default: System.out.println( "Unsupported selection!" );
25     }
26   }
27 }
```

The literal after case is a `String` literal

Example 3: SelectionByNumber

```
1  import java.util.Scanner;
2  public class SelectionByNumber {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      int choice = 0;
6      do {
7        System.out.println( "I have something to tell you." );
8        System.out.print( "Select your number (1 .. 4): " );
9        choice = console.nextInt();
10       switch ( choice ) {
11       case 1 : System.out.println( "I love Miami." ); break;
12       case 2 : System.out.println( "I love clubs." ); break;
13       case 3 : System.out.println( "I hate cold." ); break;
14       case 4 : System.out.println( "I can't dance." ); break;
15       }
16     } while ( choice >= 1 && choice <= 4 );
17   }
18 }
```

Scanner and the input given by the user

Example 3: SelectionByNumber

```
1  import java.util.Scanner;
2  public class SelectionByNumber {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      int choice = 0;
6      do {
7        System.out.println( "I have something to tell you." );
8        System.out.print( "Select your number (1 .. 4): " );
9        choice = console.nextInt();
10       switch ( choice ) {
11       case 1 : System.out.println( "I love Miami." ); break;
12       case 2 : System.out.println( "I love clubs." ); break;
13       case 3 : System.out.println( "I hate cold." ); break;
14       case 4 : System.out.println( "I can't dance." ); break;
15       }
16     } while ( choice >= 1 && choice <= 4 );
17   }
18 }
```

The loop goes until the input goes out of range 1..4

Example 3: SelectionByNumber

```
1  import java.util.Scanner;
2  public class SelectionByNumber {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      int choice = 0;
6      do {
7        System.out.println( "I have something to tell you." );
8        System.out.print( "Select your number (1 .. 4): " );
9        choice = console.nextInt();
10       switch ( choice ) {
11       case 1 : System.out.println( "I love Miami." ); break;
12       case 2 : System.out.println( "I love clubs." ); break;
13       case 3 : System.out.println( "I hate cold." ); break;
14       case 4 : System.out.println( "I can't dance." ); break;
15       }
16     } while ( choice >= 1 && choice <= 4 );
17   }
18 }
```

Receive input from the user

Example 3: SelectionByNumber

```
1  import java.util.Scanner;
2  public class SelectionByNumber {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      int choice = 0;
6      do {
7        System.out.println( "I have something to tell you." );
8        System.out.print( "Select your number (1 .. 4): " );
9        choice = console.nextInt();
10       switch ( choice ) {
11       case 1 : System.out.println( "I love Miami." ); break;
12       case 2 : System.out.println( "I love clubs." ); break;
13       case 3 : System.out.println( "I hate cold." ); break;
14       case 4 : System.out.println( "I can't dance." ); break;
15       }
16     } while ( choice >= 1 && choice <= 4 );
17   }
18 }
```

The action taken according to the input

Example 4: Throwing Dice with Switch

Throw a die a number of times specified by the user and count how many times each face shows

A switch statement when increaasing the counter

The code

```
4     Scanner console = new Scanner( System.in );
5     Random rand = new Random();
6     int numberOfRounds, die;
7     int count1 = 0, count2 = 0, count3 = 0;
8     int count4 = 0, count5 = 0, count6 = 0;
9     System.out.print( "How many throws ? " );
10    numberOfRounds = console.nextInt();
11    for ( int round = 1; round <= numberOfRounds; round ++ ) {
12      die = rand.nextInt( 6 ) + 1;
13      switch ( die ) {
14      case 1: count1 ++; break;
15      case 2: count2 ++; break;
16      case 3: count3 ++; break;
17      case 4: count4 ++; break;
18      case 5: count5 ++; break;
19      case 6: count6 ++; break;
20      }
21    }
```

The for loop

The code

```
4     Scanner console = new Scanner( System.in );
5     Random rand = new Random();
6     int numberOfRounds, die;
7     int count1 = 0, count2 = 0, count3 = 0;
8     int count4 = 0, count5 = 0, count6 = 0;
9     System.out.print( "How many throws ? " );
10    numberOfRounds = console.nextInt();
11    for ( int round = 1; round <= numberOfRounds; round ++ ) {
12      die = rand.nextInt( 6 ) + 1;
13      switch ( die ) {
14      case 1: count1 ++; break;
15      case 2: count2 ++; break;
16      case 3: count3 ++; break;
17      case 4: count4 ++; break;
18      case 5: count5 ++; break;
19      case 6: count6 ++; break;
20      }
21    }
```

Throw of a die

The code

```
4     Scanner console = new Scanner( System.in );
5     Random rand = new Random();
6     int numberOfRounds, die;
7     int count1 = 0, count2 = 0, count3 = 0;
8     int count4 = 0, count5 = 0, count6 = 0;
9     System.out.print( "How many throws ? " );
10    numberOfRounds = console.nextInt();
11    for ( int round = 1; round <= numberOfRounds; round ++ ) {
12      die = rand.nextInt( 6 ) + 1;
13      switch ( die ) {
14      case 1: count1 ++; break;
15      case 2: count2 ++; break;
16      case 3: count3 ++; break;
17      case 4: count4 ++; break;
18      case 5: count5 ++; break;
19      case 6: count6 ++; break;
20      }
21    }
```

The switch

The End