```java
package club.westcs.OOPNotes;

import java.util.Random;
import java.util.Scanner;

public class Viking {

    //Attributes
        /*
         * the parts of an object
         * declare the parts but do not assign them values example   private String name;
         * set to private access modifier
         *      private
         *          an access modifier that creates a variable available only to this class
         */
        private String name;
        private boolean alive;
        private int weapons;
        private Random rand;
        private Scanner scan;

    //Constructor
        /*
         * gives values to all attributes for each instance of the object
         * (Puts the object together)
         * runs automatically when you make a new instance of an object
         * Always set to public
         * does not have a return type (no void String....)
         * Always named and spelled just like the class
         */
        public Viking() {
            rand = new Random();
            scan = new Scanner(System.in);
            alive = true;
            weapons = rand.nextInt(4) + 2; // randomly 2 - 5 weapons
            name = setName();
            System.out.println(this.toString());
        }                                                   // Viking v = new Viking();
```

```
39
40      public Viking(String name) { //overloaded constructor (if a new instance has a String it will call this one)
41          rand = new Random();
42          scan = new Scanner(System.in);
43          alive = true;
44          weapons = rand.nextInt(4) + 2; // randomly 2 - 5 weapons
45          this.name = name;
46          System.out.println(this.toString());
47      }                                                   //Viking v = new Viking("Bob");
48
49  //Methods
50      /*
51       * The stuff an object can do.
52       * Look like functions but belong to objects.
53       * Can be public or private
54       * Used to get and set values of attributes.
55       */
56  public String setName() {
57      System.out.println("What is this Viking's name?");
58      return scan.nextLine();
59  }
60
61  public String toString() {
62      return this.name + " has " + this.weapons + " weapons.";
63  }
64
65  public boolean isAlive() {
66      return this.alive;
67  }
68
69  public void setAlive() {
70      if(this.alive && this.weapons <= 0) {
71          System.out.println(this.name + " has fallen and gone to Valhalla.");
72          this.alive = false;
73      }
74  }
75
76      public void loseAWeapon() {
77          System.out.println("Garrrrrhhhh");
78          System.out.println(this.name + " has lost a weapon.");
79          this.weapons--;
80          setAlive();
81      }
82
83      public void attack(Viking other) {
84          System.out.println(this.name + " has attacked " + other.getName() + ".");
85          if(other.isAlive() && rand.nextBoolean()) {
86              other.loseAWeapon();
87          }
88      }
89
90      public String getName() {
91          return this.name;
92      }
93  }
94
```