```
1  /**
2   * This program runs a simulation for 100 years. This
   simulation predicts the affect of beetles, drought, and
   fires on
3   * trees of different ages.The program is also capable of
   determining the effects of seeding clouds and spraying
4   * pesticides of the crops.
5   */
6
7  import java.util.concurrent.ThreadLocalRandom;
8  public class NorthwestLumber {
9      public static void main(String args []){
10         Trees start = new Trees(400000, 300000,200000,
   100000);//Starting crop
11
12         for(int i = 0; i<=100; i++){//For 100 years
13             start.rainFall();//calculate rainfall
14             start.seedingClouds();//seed clouds
15             start.surviveWeather();//See which tress
   survive rainfall
16             start.Plant();//replace dead
17             start.surviveBeetles();//Survive beetles
18             start.Plant();//replace dead
19             start.surviveFire();//Survive fire
20             start.Harvest();//harvest five year old plants
21             start.Age();//age up plants
22             start.Plant();//replace dead and harvested
23             if(90<=i){//Print last ten years
24                 System.out.println("After "+ i+" years we
   will have "+ start.one +" one year old trees, "+start.two+
   " two year old trees, "+start.three+" three year old trees
   , and "+start.four+" year old trees.");
25             }
26         }//end of 100 year loop
27     }//End of main
28 }//End of NorthwestLumber
29 class Trees {//Start of tree class
30     private Boolean spray;//If trees are sprayed
31     double one;//one year old trees
32     double two;//two year old trees
33     double three;//three year old trees
34     double four;//four year old trees
35     double five = 0;//Trees ready to harvest
36
37     private int rain;//1 = Drought 2=Moderate 3=Heavy
```

```
38      public Trees(int a, int b, int c, int d){//Tree
    constructor
39          one = a;
40          two = b;
41          three = c;
42          four = d;
43
44      }//end of constructor
45      public void rainFall()throws NullPointerException{//
    Method calculates Rainfall
46          int randomNum = ThreadLocalRandom.current().
    nextInt(1, 11 + 1);//Random value from 1-11
47          if(randomNum <= 3){//If less than 3 inch it is a
    drought
48              rain= 1;
49          }
50          else if(randomNum <= 10){//more than three less
    than 11 is moderate rainfall
51              rain = 2;
52          }
53          else{//Any rainfall over 10 is heavy rainfall
54              rain = 3;
55          }
56      }//End of Rainfall
57      public void seedingClouds(){//Cloud seeding
58          int randomNum = ThreadLocalRandom.current().
    nextInt(1, 100 + 1);
59          if(rain == 1){//Seeding for drought
60              if(randomNum<= 89){//89% chance of seeding
61                  rain = 2;
62                  return;
63              }
64          }
65          else if(rain == 2){//Seeding for moderate
66              if(randomNum<=18){//18% chance
67                  rain = 3;
68              }
69          }
70      }//end of seeding
71      public void surviveWeather(){//Start of survive
    weather
72          if(rain == 1){//Survive Drought
73             one = (one - (one*0.1));
74             two = (two - (two*0.1));
75             three = (three- (three*0.3));
```

```
76              four = (four - (four*0.35));
77          }
78          else if(rain == 2){//Survive Moderate rainfall
79              one = (one - (one*0.05));
80              two = (two - (two*0.05));
81              three = (three- (three*0.05));
82              four = (four - (four*0.05));
83          }
84          else{//survive Heavy rainfall
85              one = (one - (one*0.02));
86              two = (two - (two*0.03));
87              three = (three- (three*0.03));
88              four = (four - (four*0.04));
89          }
90      }//end of surviveWeather
91      public void surviveBeetles() {//start of survive
    beetles
92          if (spray == Boolean.TRUE) {//Check if plants are
     sprayed
93              if (rain == 1) {//Drought sprayed plant
    survival
94                  one = (one - (one * 0.1));
95                  two = (two - (two * 0.15));
96                  three = (three - ((three/2) * 0.3));
97                  four = (four - ((four/2) * 0.3));
98              } else if (rain == 2) {//Moderate sprayed
    plant survival
99                  one = (one - (one * 0.05));
100                 two = (two - (two * 0.05));
101                 three = (three - ((three/2) * 0.1));
102                 four = (four - ((four/2) * 0.1));
103             } else {//Heavy sprayed plant survival
104                 three = (three - ((three/2) * 0.02));
105                 four = (four - ((four/2) * 0.02));
106             }
107
108         }//end of sprayed trees
109         else {//Survival of non-sprayed trees
110             if (rain == 1) {//Drought beetle survival
111                 one = (one - (one * 0.1));
112                 two = (two - (two * 0.15));
113                 three = (three - (three * 0.3));
114                 four = (four - (four * 0.3));
115             } else if (rain == 2) {//Moderate beetle
    survival
```

```
116                 one = (one - (one * 0.05));
117                 two = (two - (two * 0.05));
118                 three = (three - (three * 0.1));
119                 four = (four - (four * 0.1));
120             } else {//Heavy beetle survival
121                 three = (three - (three * 0.02));
122                 four = (four - (four * 0.02));
123             }
124         }//end of non-sprayed trees
125     }//end of survive beetles
126     public void surviveFire(){//Start of survive fire
127         if(rain == 1){//Fire survival Drought
128             one = (one - (one*0.15));
129             two = (two - (two*0.18));
130             three = (three- (three*0.22));
131             four = (four - (four*0.3));
132         }
133         else if(rain == 2){//Fire survival Moderate
134             one = (one - (one*0.1));
135             two = (two - (two*0.12));
136             three = (three- (three*0.15));
137             four = (four - (four*0.2));
138         }
139         else{//Fire survival Heavy
140             one = (one - (one*0.02));
141             two = (two - (two*0.07));
142             three = (three- (three*0.1));
143             four = (four - (four*0.15));
144         }
145 
146     }//End of surviveFire
147     public double getTotal(){//Start of get total
148         double total = one + two + three + four;//add
    total number of trees
149         return total;
150     }//end of getTotal
151     public void Harvest(){//Start of harvest
152         one = one + five;
153         five = 0;
154     }//End of Harvest
155     public void Age(){//Start of Age
156         if(rain == 1){//growing in a drought
157             five=(four*0.65);
158             four = (four * 0.35)+(three*0.4);
159             three = (three * 0.6)+(two*0.9);
```

```
160             two = (two*0.1)+(one*0.9);
161             one = (one * 0.1);
162 
163         }
164         else if (rain == 2){//growing in moderate
    rainfall
165             five=(four*0.99);
166             four = (four * 0.01)+(three*0.98);
167             three = (three * 0.01)+(two*0.98);
168             two = (two*0.01)+(one*0.99);
169             one = (one * 0.01);
170 
171         }
172         else {//growing in heavy rainfall
173             five=(four*0.96);
174             four = (four * 0.04)+(three*0.97);
175             three = (three * 0.03)+(two*0.97);
176             two = (two*0.03)+(one*0.98);
177             one = (one * 0.02);
178 
179         }
180     }//end of age
181     public void Plant(){//Start of plant
182         double total = getTotal();//total plants on lot
183         if(total < 1000000){//if total is less than a
    million
184             one = one + (1000000-total);//plant free
    space
185         }
186     }//end of plant
187 }//End of tree class
188 
```