```
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.surviveWeather();//See which tress
   survive rainfall
15             start.Plant();//replace dead
16             start.surviveBeetles();//Survive beetles
17             start.Plant();//replace dead
18             start.surviveFire();//Survive fire
19             start.Harvest();//harvest five year old plants
20             start.Age();//age up plants
21             start.Plant();//replace dead and harvested
22             if(90<=i){//Print last ten years
23                 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.");
24             }
25         }//end of 100 year loop
26     }//End of main
27 }//End of NorthwestLumber
28 class Trees {//Start of tree class
29     private Boolean spray= Boolean.TRUE;//If trees are
   sprayed
30     double one;//one year old trees
31     double two;//two year old trees
32     double three;//three year old trees
33     double four;//four year old trees
34     double five = 0;//Trees ready to harvest
35
36     private int rain;//1 = Drought 2=Moderate 3=Heavy
```

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

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

```
115             }
116         }//end of non-sprayed trees
117     }//end of survive beetles
118     public void surviveFire(){//Start of survive fire
119         if(rain == 1){//Fire survival Drought
120             one = (one - (one*0.15));
121             two = (two - (two*0.18));
122             three = (three- (three*0.22));
123             four = (four - (four*0.3));
124         }
125         else if(rain == 2){//Fire survival Moderate
126             one = (one - (one*0.1));
127             two = (two - (two*0.12));
128             three = (three- (three*0.15));
129             four = (four - (four*0.2));
130         }
131         else{//Fire survival Heavy
132             one = (one - (one*0.02));
133             two = (two - (two*0.07));
134             three = (three- (three*0.1));
135             four = (four - (four*0.15));
136         }
137 
138     }//End of surviveFire
139     public double getTotal(){//Start of get total
140         double total = one + two + three + four;//add
    total number of trees
141         return total;
142     }//end of getTotal
143     public void Harvest(){//Start of harvest
144         one = one + five;
145         five = 0;
146     }//End of Harvest
147     public void Age(){//Start of Age
148         if(rain == 1){//growing in a drought
149             five=(four*0.65);
150             four = (four * 0.35)+(three*0.4);
151             three = (three * 0.6)+(two*0.9);
152             two = (two*0.1)+(one*0.9);
153             one = (one * 0.1);
154 
155         }
156         else if (rain == 2){//growing in moderate
    rainfall
157             five=(four*0.99);
```

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