

```

1  //
2  //  SIEVE. The Sieve of Eratosthenes.
3  //
4  //    James B. Moen
5  //    08 Oct 19
6  //
7  //  Test the SIEVE class, for 30 points total.
8  //
9
10 //
11 class Sieve{
12     public boolean[] numbers;
13     Sieve(int max) {
14         numbers = new boolean[max];
15
16         if (max < 2) {
17             throw new IllegalArgumentException();
18         }
19         for(int i = 0; i<max; i++){
20             if (i == 0 || i == 1) {
21                 numbers[i] = false;
22             }
23             else {
24                 numbers[i] = true;
25             }
26         }
27     }
28
29     public void findPrimes(){
30         for(int i = 0; i<numbers.length; i++) {
31             if (numbers[i]) {
32                 for(int j = 2*i; j<numbers.length; j = j + i) {
33                     numbers[j] = false;
34                 }
35             }
36         }
37     }
38
39     public String toString(){
40         String myString = "";
41         for(int i = 0; i < numbers.length; i++) {
42             if(numbers[i]) {
43                 myString = myString + i + " ";
44             }
45         }
46         return myString;
47     }

```

```

48
49 }
50 //
51
52 // TOSS THE KNEES. Run SIEVE on some examples.
53
54 class TossTheKnees
55 {
56
57 // MAIN. Find some primes.
58
59 public static void main(String [] args)
60 {
61     Sieve sieve = null; // We must initialize SIEVE or Java will
    • cry.
62
63 // 5 points. This must print "Sieve size must be at least 2." but
    • without the
64 // quotes.
65
66     try
67     {
68         sieve = new Sieve(0);
69     }
70     catch (IllegalArgumentException oops)
71     {
72         System.out.println("Sieve size must be at least 2.");
73     }
74
75 // 5 points. This must print nothing.
76
77     try
78     {
79         sieve = new Sieve(100);
80     }
81     catch (IllegalArgumentException oops)
82     {
83         System.out.println("Sieve size must be at least 2.");
84     }
85
86 // 10 points. This must print integers from 2 to 99, separated by
    • blanks.
87
88     System.out.println(sieve);
89
90 // 10 points. This must print the prime numbers between 2 and 99,
    • separated by

```

```
    • separated by  
91 // blanks. They are:  
92 //  
93 // 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83  
    • 89 97  
94  
95     sieve.findPrimes();  
96     System.out.println(sieve);  
97 }  
98 }  
99
```