3.6 Problem Solving: Test Cases

Each branch of your program should be covered by a test case.

Consider how to test the tax computation program from Section 3.4. Of course, you cannot try out all possible inputs of marital status and income level. Even if you could, there would be no point in trying them all. If the program correctly computes one or two tax amounts in a given bracket, then we have a good reason to believe that all amounts will be correct.

You want to aim for complete *coverage* of all decision points. Here is a plan for obtaining a comprehensive set of test cases:

- There are two possibilities for the marital status and two tax brackets for each status, yielding four test cases.
- Test a handful of *boundary* conditions, such as an income that is at the boundary between two brackets, and a zero income.
- If you are responsible for error checking (which is discussed in Section 3.8), also test an invalid input, such as a negative income.

Make a list of the test cases and the expected outputs:

Test Case	Expected Output	Comment
30,000 s	3,000	10% bracket
72,000 s	13,200	3,200 + 25% of 40,000
50,000 m	5,000	10% bracket
104,000 m	16,400	6,400 + 25% of 40,000
32,000 s	3,200	boundary case
0	0	boundary case

It is a good idea to design test cases before implementing a program.

When you develop a set of test cases, it is helpful to have a flowchart of your program (see Section 3.5). Check off each branch that has a test case. Include test cases for the boundary cases of each decision. For example, if a decision checks whether an input is less than 100, test with an input of 100.

It is always a good idea to design test cases *before* starting to code. Working through the test cases gives you a better understanding of the algorithm that you are about to implement.

■■ **1.** Consider this code segment for computing income tax, then complete the table below.

```
double income = in.nextDouble();
String maritalStatus = in.next();
if (maritalStatus.equals("s"))  // Condition 1
{
   if (income <= 30000) // Condition 2
   {
      tax = 0.10 * income; // Branch 1
   }
   else
   {
      tax = 3000 + 0.25 * (income - 30000); // Branch 2
   }
}
else
{
   if (income <= 60000) // Condition 3
   {
      tax = 0.10 * income; // Branch 3
   }
   else
   {
      tax = 6000 + 0.25 * (income - 60000); // Branch 4
   }
}
```

Press start to begin.

Start

Which branch is tested by the input `40000 s` ?	
Which branch is tested by the input `70000 m`?	

Which branch is tested by the input `30000 s`?	
Which branch is not tested by any of these test cases?	
Provide a test case for that branch.	
Give a boundary test case for condition 3.	

- **2.** The program segment below has been tested by giving variable `number` the values 777, 1000, and 1035. What additional value should be used in order to achieve coverage of all decision points?

```
if (number > 1000)
{
   if (number % 2 == 0)
   {
      System.out.print("A large even number");
   }
   else
   {
      System.out.print("A large odd number");
   }
}
else
{
   if (number % 2 == 0)
   {
```

```
            System.out.print("A small even number");
         }
         else
         {
            System.out.print("A small odd number");
         }
      }
```

- ☐ 7
- ☐ 778
- ☐ 1005
- ☐ 1006

27. Using Figure 1 as a guide, follow the process described in Section 3.6 to design a set of test cases for the `ElevatorSimulation.java` program in Section 3.1.

Show Answer

28. What is a boundary test case for the algorithm in How To 3.1? What is the expected output?

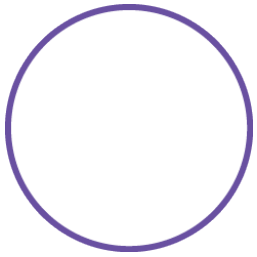
Show Answer

29. Using Figure 3 as a guide, follow the process described in Section 3.6 to design a set of test cases for the `EarthquakeStrength.java` program in Section 3.3.

Show Answer

30. Suppose you are designing a part of a program for a medical robot that has a sensor returning an *x*- and *y*-location (measured in cm). You need to check whether the sensor location is inside the circle, outside the circle, or on the boundary (specifically, having a distance of less than

1 mm from the boundary). Assume the circle has center (0, 0) and a radius of 2 cm. Give a set of test cases.



Show Answer

PRACTICE IT Now you can try these exercises at the end of the chapter: •• R3.16, •• R3.17.