Introduction to Programming in C++ Seventh Edition

Chapter 2: Beginning the Problem-Solving Process

Chapter Objectives

- Explain the problem-solving process used to create a computer program
- Analyze a problem
- Complete an IPO chart
- Plan an algorithm using pseudocode and flowcharts
- Desk-check an algorithm

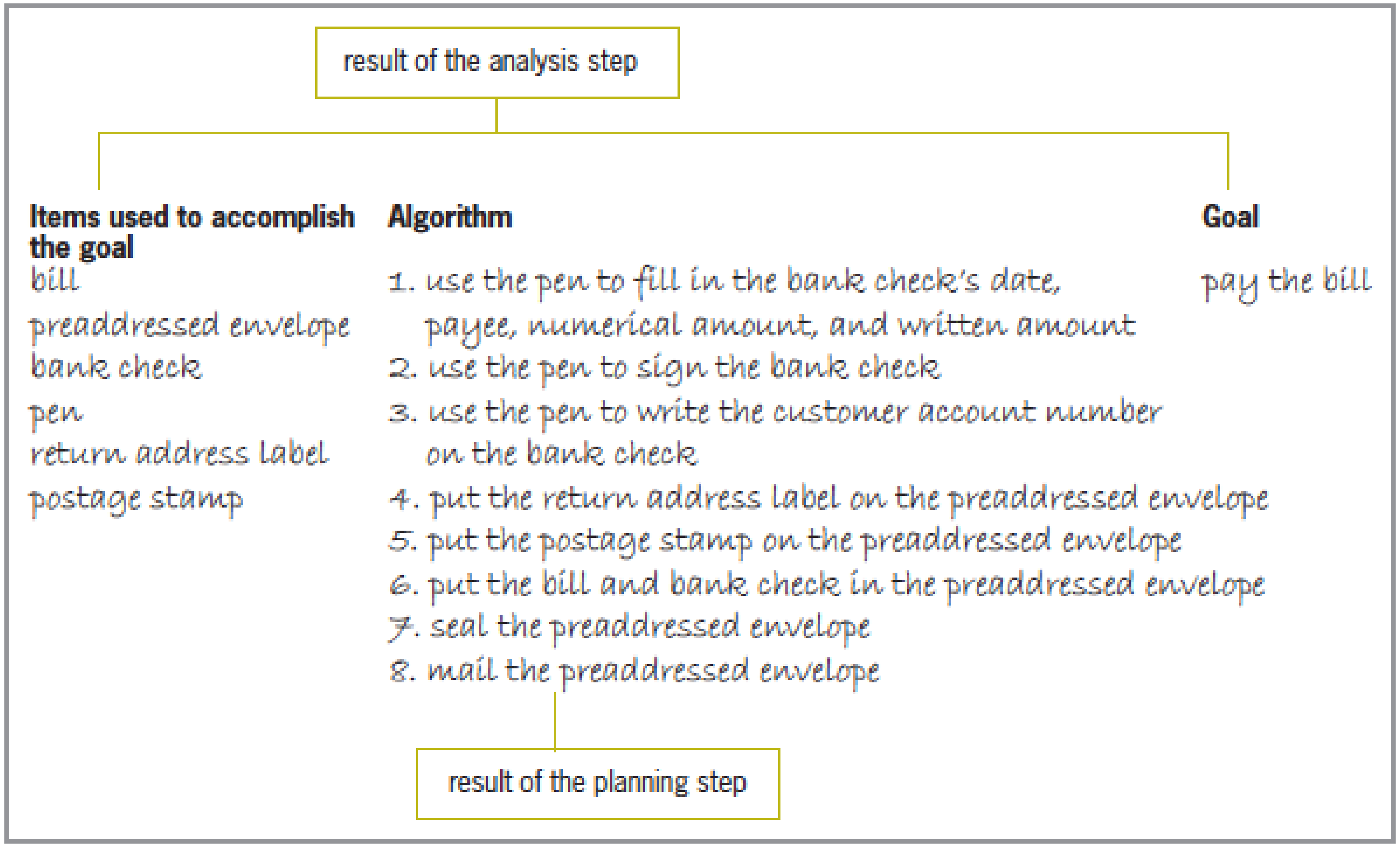
Problem Solving

- People solve hundreds of simple problems every day without thinking about how they do it
- Understanding the thought process involved can help in solving more complex problems
- You can also use a similar process to design a computer solution to a problem (computer program)

Solving Everyday Problems

- First step in solving a problem: analyze it
  - Example: paying and mailing a bill
- Next, you plan, review, implement, and evaluate the solution
- After this, it may be necessary to modify the solution

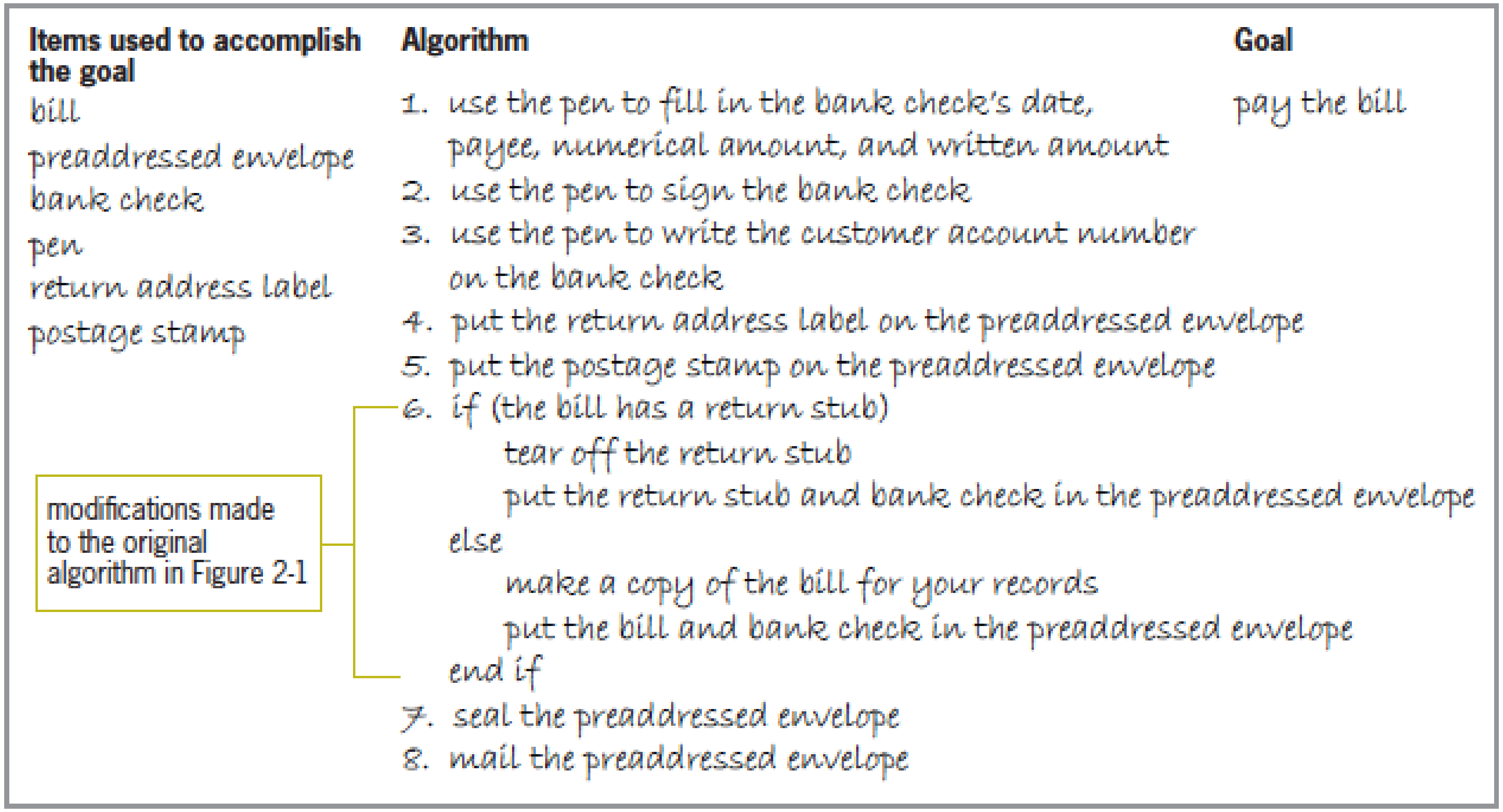
Solving Everyday Problems (cont'd.)



Items used to accomplish the goal	Algorithm	Goal
bill	1. use the pen to fill in the bank check's date, payee, numerical amount, and written amount	pay the bill
preaddressed envelope	2. use the pen to sign the bank check	
bank check	3. use the pen to write the customer account number on the bank check	
pen	4. put the return address label on the preaddressed envelope	
return address label	5. put the postage stamp on the preaddressed envelope	
postage stamp	6. put the bill and bank check in the preaddressed envelope	
	7. seal the preaddressed envelope	
	8. mail the preaddressed envelope	

Figure 2-1 Summary of the analysis and planning steps for the bill paying problem

Solving Everyday Problems (cont'd.)



Items used to accomplish the goal
- bill
- preaddressed envelope
- bank check
- pen
- return address label
- postage stamp

Algorithm

```
1. use the pen to fill in the bank check's date,
   payee, numerical amount, and written amount
2. use the pen to sign the bank check
3. use the pen to write the customer account number
   on the bank check
4. put the return address label on the preaddressed envelope
5. put the postage stamp on the preaddressed envelope
6. if (the bill has a return stub)
       tear off the return stub
       put the return stub and bank check in the preaddressed envelope
   else
       make a copy of the bill for your records
       put the bill and bank check in the preaddressed envelope
   end if
7. seal the preaddressed envelope
8. mail the preaddressed envelope
```

modifications made to the original algorithm in Figure 2-1

Goal

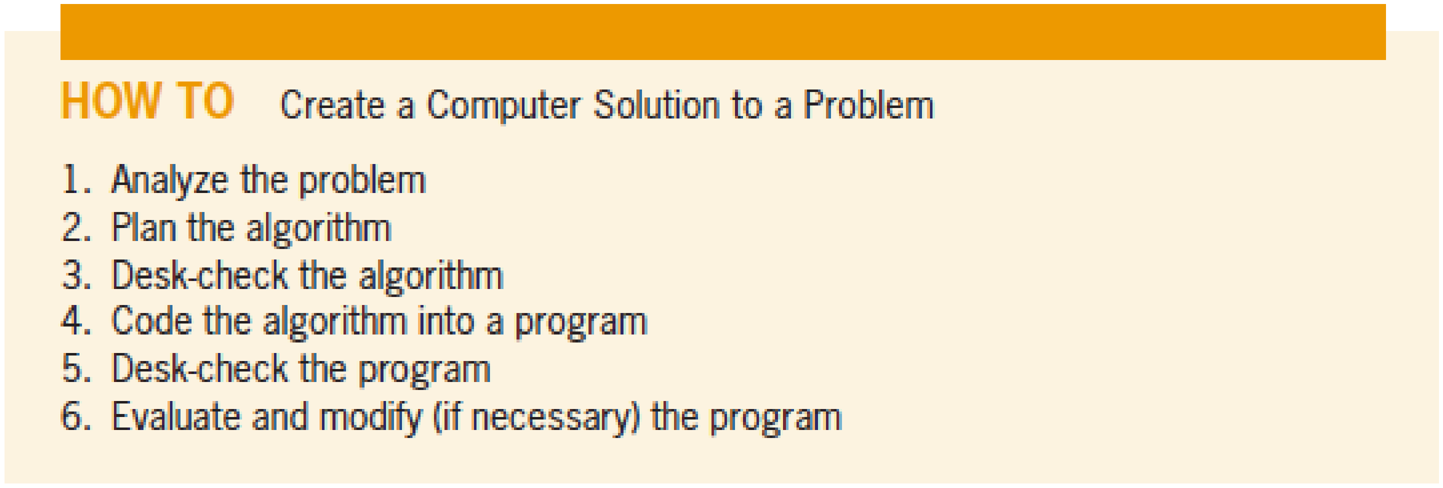
pay the bill

Figure 2-2 Modified algorithm for the bill paying problem

Creating Computer Solutions to Problems

- A similar process to everyday problem solving is used to create computer programs
- A computer program is a solution implemented on a computer
- There are six steps to creating a computer solution to a problem

Creating Computer Solutions to Problems (cont'd.)



HOW TO Create a Computer Solution to a Problem

1. Analyze the problem
2. Plan the algorithm
3. Desk-check the algorithm
4. Code the algorithm into a program
5. Desk-check the program
6. Evaluate and modify (if necessary) the program

Figure 2-3 How to create a computer solution to a problem

Step 1–Analyzing the Problem

- It is essential to understand a problem before creating a solution to it
- Analyze a problem to:
  - Determine the goal of solving it (**Output**)
  - Determine the items needed to achieve that goal (**Input**)
- Always search first for the output

Step 1—Analyzing the Problem (cont'd.)

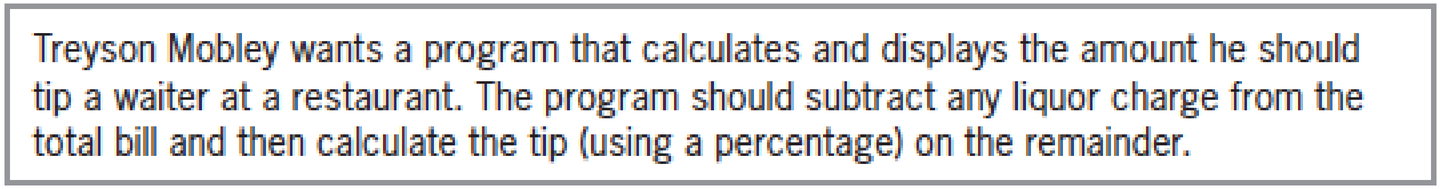
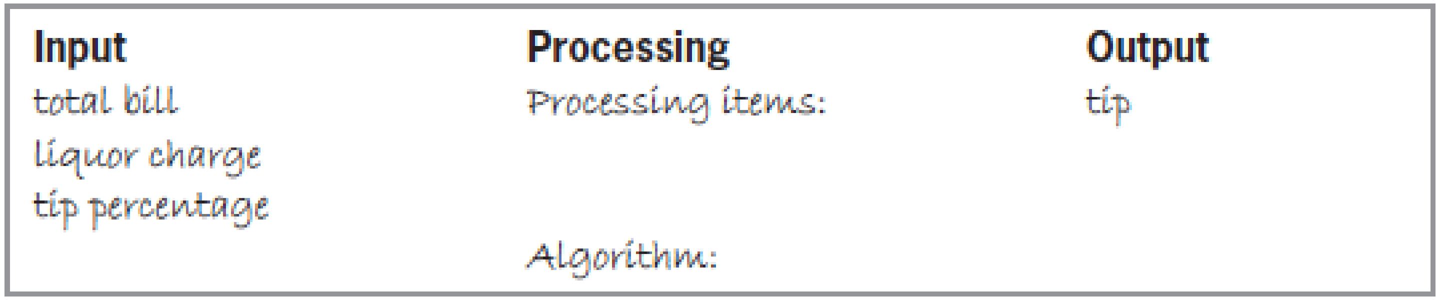

Treyson Mobley wants a program that calculates and displays the amount he should tip a waiter at a restaurant. The program should subtract any liquor charge from the total bill and then calculate the tip (using a percentage) on the remainder.

Figure 2-4 Problem specification for Treyson Mobley

Step 1–Analyzing the Problem (cont'd.)

- Some programmers use an **IPO chart** to organize and summarize the results of a problem analysis
  - **IPO**: Input, processing, and output

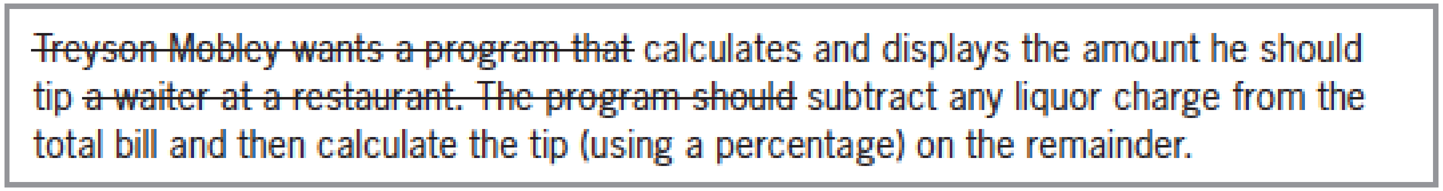


Input	Processing	Output
total bill	Processing items:	tip
liquor charge		
tip percentage		
	Algorithm:	

Figure 2-5 Partially completed IPO chart showing the input and output items

Hints for Analyzing Problems

- Several readings of the problem may be necessary to fully understand the problem
- Cross out irrelevant information in the problem description

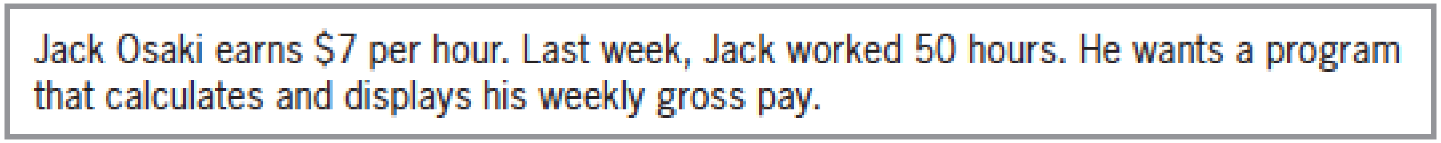


~~Treyson Mobley wants a program that~~ calculates and displays the amount he should tip ~~a waiter at a restaurant. The program should~~ subtract any liquor charge from the total bill and then calculate the tip (using a percentage) on the remainder.

Figure 2-6 Problem specification with unimportant information crossed out

Hints for Analyzing Problems (cont'd.)

- Some problem specifications contain incomplete information



Jack Osaki earns $7 per hour. Last week, Jack worked 50 hours. He wants a program that calculates and displays his weekly gross pay.

Figure 2-7 Problem specification that does not contain enough information

Hints for Analyzing Problems (cont'd.)

- Distinguish between information that is missing and information that is implied

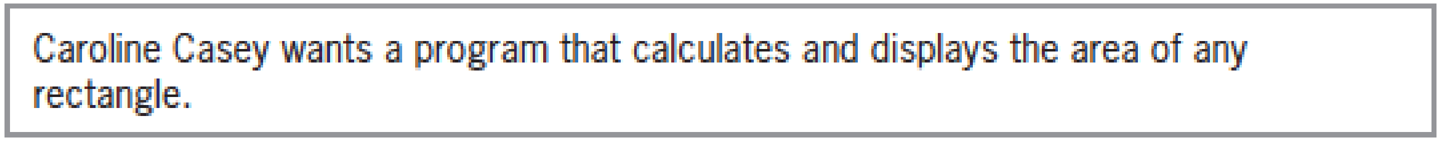
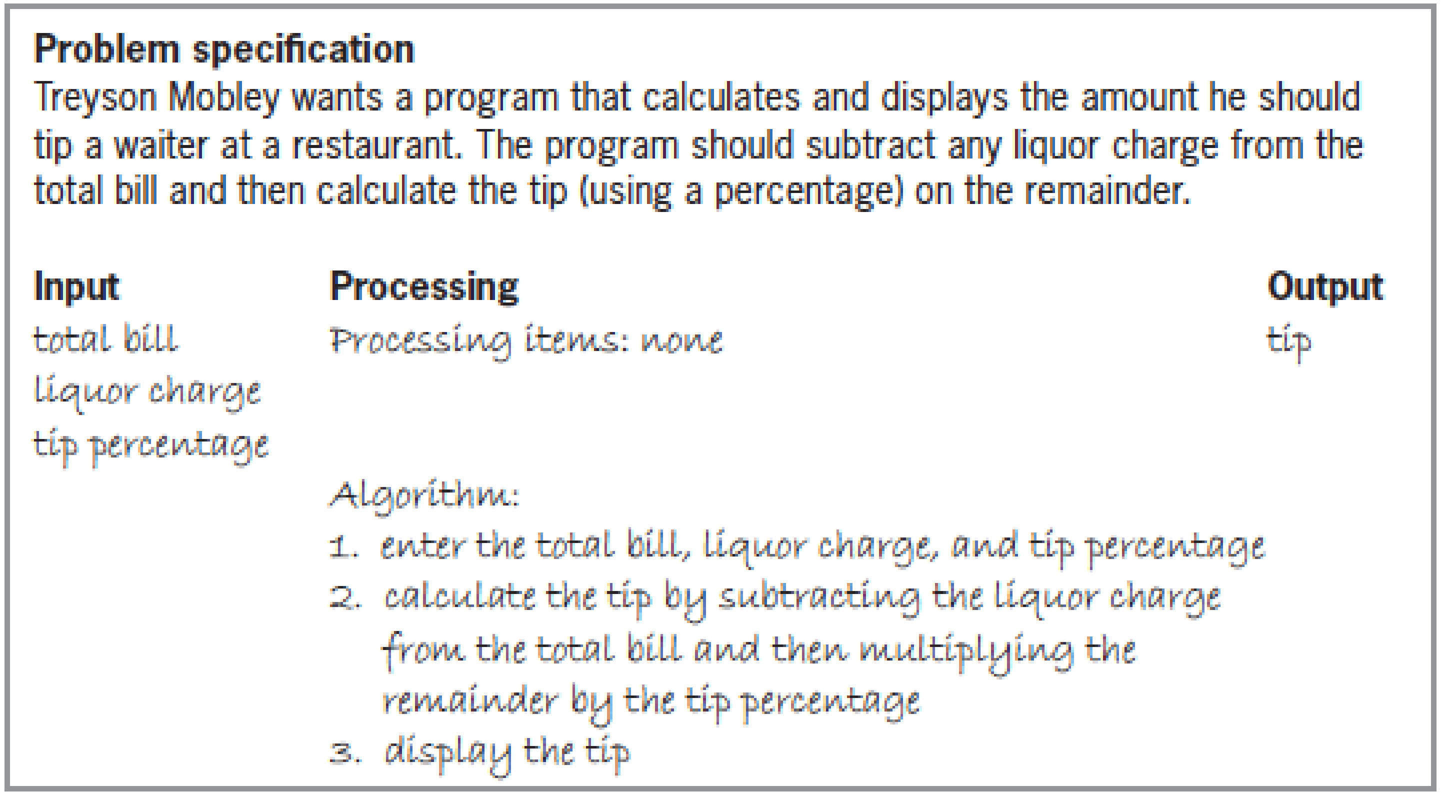

Caroline Casey wants a program that calculates and displays the area of any rectangle.

Figure 2-8 Problem specification in which the input is not explicitly stated

Step 2–Planning the Algorithm

- **Algorithm**: set of instructions that will transform the problem’s input into its output
  - Record in the Processing column of the IPO chart
  - Can be written as pseudocode or a flowchart
- **Pseudocode**: tool programmers use to help plan an algorithm
  - Short English statements
  - Not standardized
  - Not understandable by a computer

Step 2–Planning the Algorithm (cont'd.)



Problem specification

Treyson Mobley wants a program that calculates and displays the amount he should tip a waiter at a restaurant. The program should subtract any liquor charge from the total bill and then calculate the tip (using a percentage) on the remainder.

Input	Processing	Output
total bill liquor charge tip percentage	Processing items: none	tip

```
Algorithm:
1. enter the total bill, liquor charge, and tip percentage
2. calculate the tip by subtracting the liquor charge
   from the total bill and then multiplying the
   remainder by the tip percentage
3. display the tip
```

Figure 2-9 Problem specification and IPO chart for the Treyson Mobley problem

Step 2–Planning the Algorithm (cont'd.)

- **Flowcharts** are also used to plan an algorithm
  - Use standardized symbols
  - Symbols connected with **flowlines**
  - Oval: **start/stop symbol**
    - Represents beginning and end of algorithm
  - Rectangle: **process symbol**
    - Represents tasks such as calculations
  - Parallelogram: **input/output symbol**
    - Represents I/O tasks

Step 2–Planning the Algorithm (cont'd.)

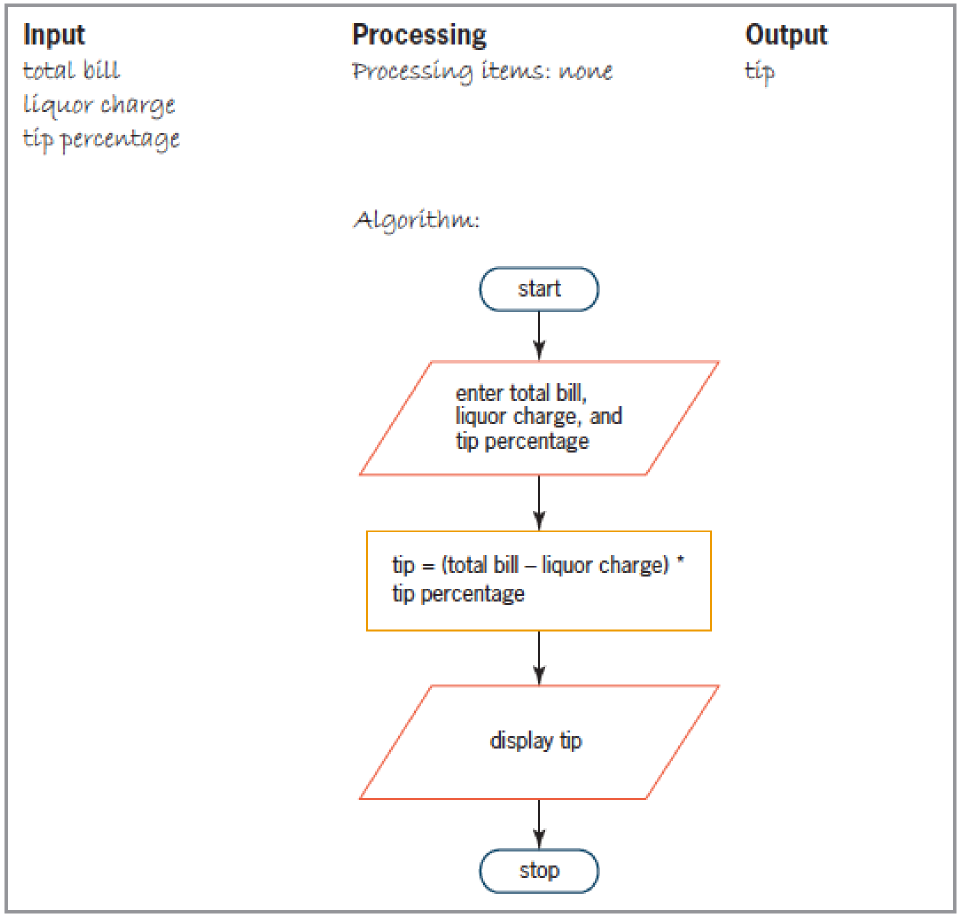
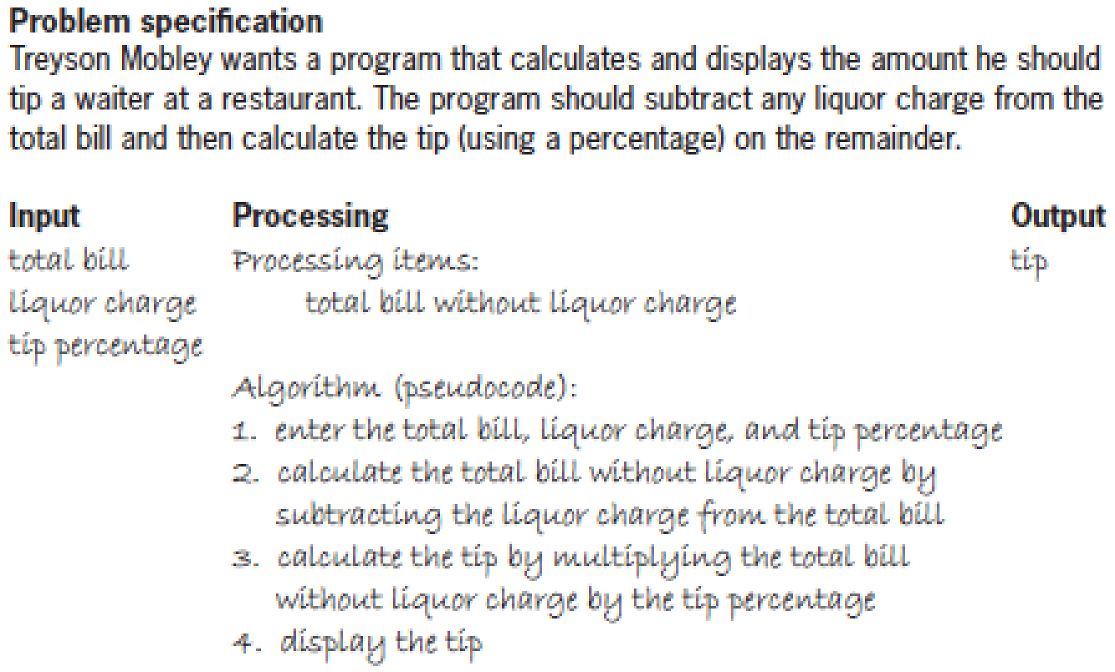


Figure 2-10 Figure 2-9's algorithm in flowchart form

Step 2–Planning the Algorithm (cont'd.)

- A problem can have more than one solution



Problem specification
Treyson Mobley wants a program that calculates and displays the amount he should tip a waiter at a restaurant. The program should subtract any liquor charge from the total bill and then calculate the tip (using a percentage) on the remainder.

Input	Processing	Output
total bill	Processing items:	tip
liquor charge	total bill without liquor charge	
tip percentage		

```
Algorithm (pseudocode):
1. enter the total bill, liquor charge, and tip percentage
2. calculate the total bill without liquor charge by
   subtracting the liquor charge from the total bill
3. calculate the tip by multiplying the total bill
   without liquor charge by the tip percentage
4. display the tip
```

Figure 2-11 A different solution to the Treyson Mobley problem (pseudocode)

Step 2–Planning the Algorithm (cont'd.)

- Processing item: an intermediate value (neither input nor output) the algorithm uses to transform input into output

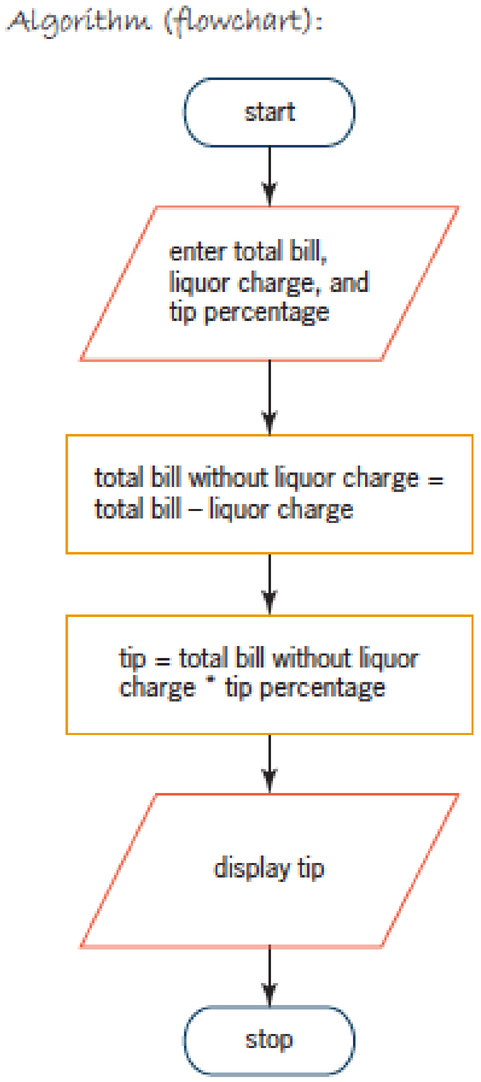

Algorithm (flowchart):
start
enter total bill, liquor charge, and tip percentage
total bill without liquor charge = total bill – liquor charge
tip = total bill without liquor charge * tip percentage
display tip
stop

Figure 2-11 A different solution to the Treyson Mobley problem (flowchart)

Step 3–Desk-Checking the Algorithm

- **Desk-checking** an algorithm verifies that it is correct
  - Refers to checking an algorithm by hand, rather than with a computer
  - Also called **hand-tracing**
- Choose sample data and manually compute the expected output value
- Creating a desk-check table can be helpful

Step 3—Desk-Checking the Algorithm (cont'd.)

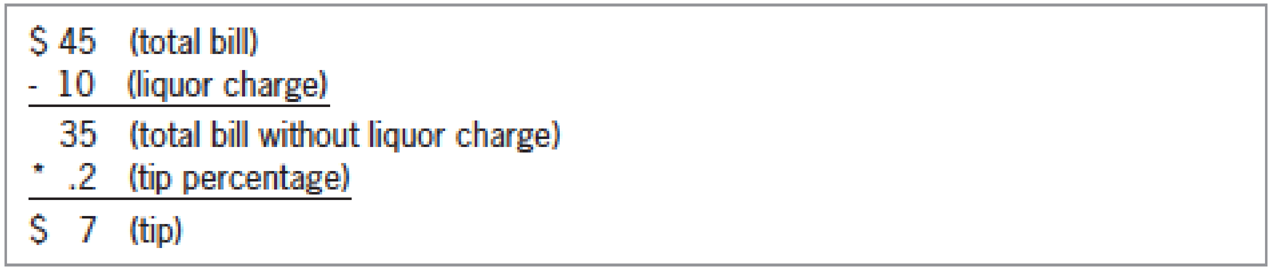
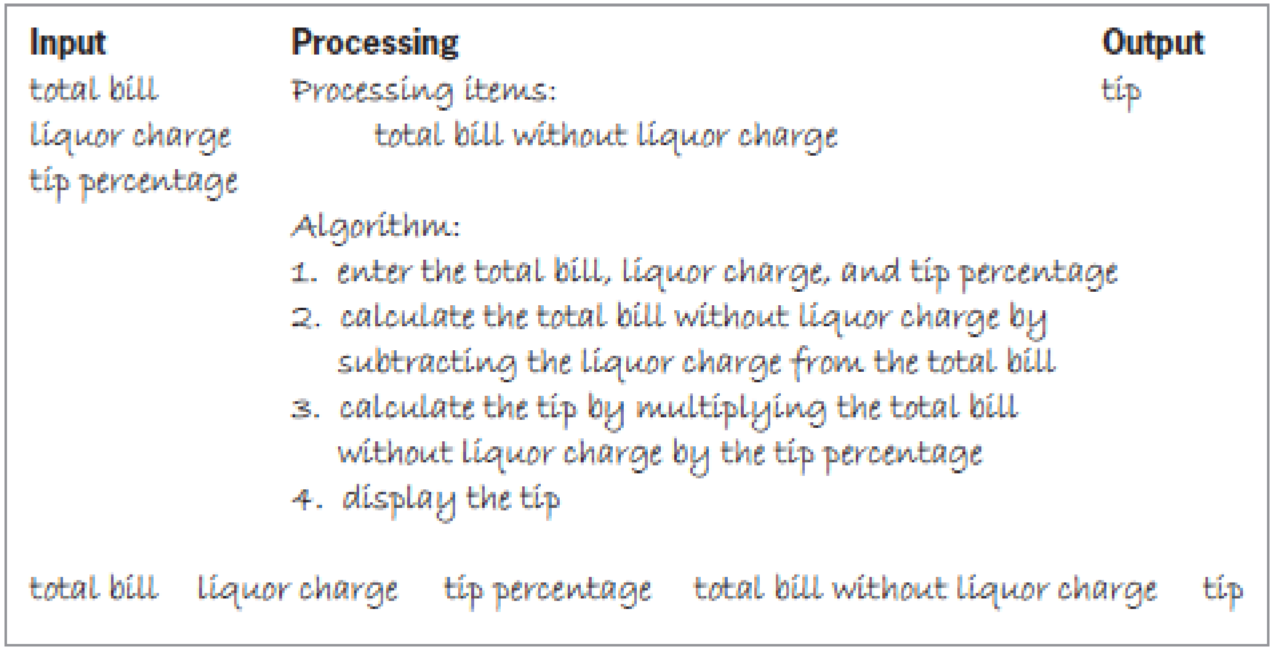

```
$ 45   (total bill)
- 10   (liquor charge)
-------------------------
  35   (total bill without liquor charge)
*  .2  (tip percentage)
-------------------------
$  7   (tip)
```

Figure 2-12 Manual tip calculation for the first desk-check

Step 3–Desk-Checking the Algorithm (cont'd.)

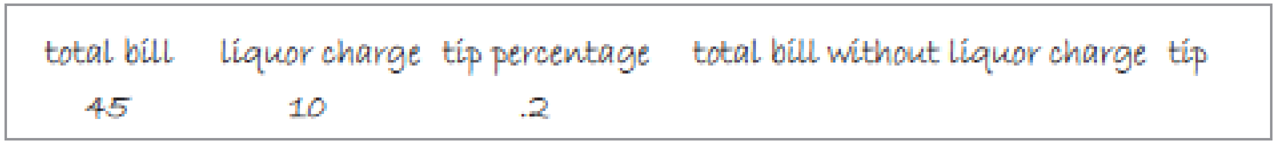


Input	Processing	Output
total bill liquor charge tip percentage	Processing items: total bill without liquor charge Algorithm: 1. enter the total bill, liquor charge, and tip percentage 2. calculate the total bill without liquor charge by subtracting the liquor charge from the total bill 3. calculate the tip by multiplying the total bill without liquor charge by the tip percentage 4. display the tip	tip

total bill	liquor charge	tip percentage	total bill without liquor charge	tip

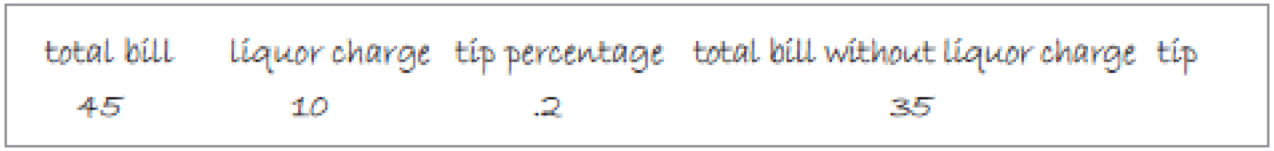
Figure 2-13 Treyson Mobley solution and partially completed desk-check table

Step 3–Desk-Checking the Algorithm (cont'd.)



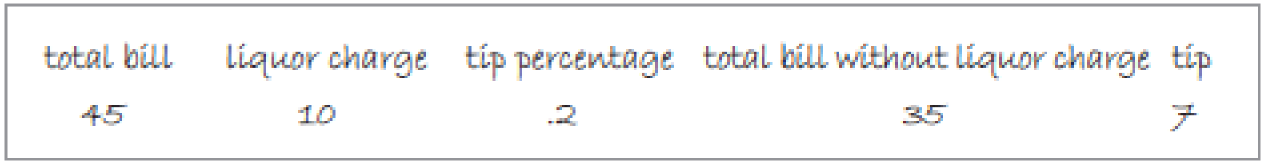
total bill	liquor charge	tip percentage	total bill without liquor charge	tip
45	10	.2		

Figure 2-14 Input values entered in the desk-check table



total bill	liquor charge	tip percentage	total bill without liquor charge	tip
45	10	.2	35	

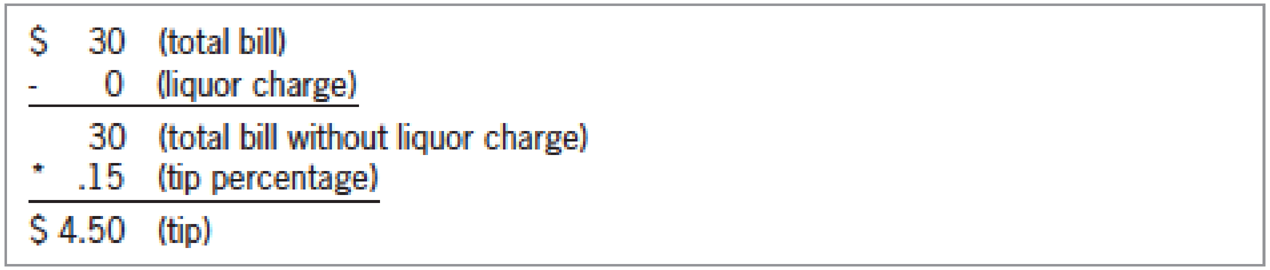
Figure 2-15 Processing item's value entered in the desk-check table



total bill	liquor charge	tip percentage	total bill without liquor charge	tip
45	10	.2	35	7

Figure 2-16 Output value entered in the desk-check table

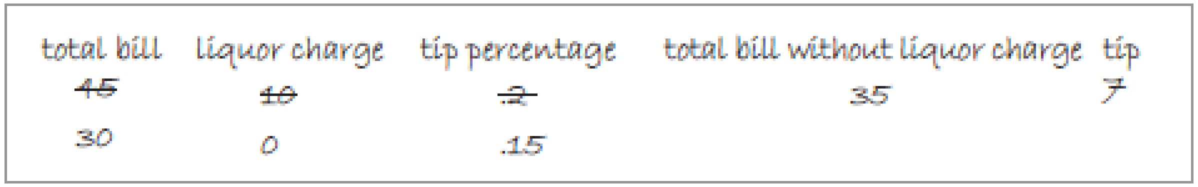
Step 3–Desk-Checking the Algorithm (cont'd.)



```
$    30   (total bill)
-     0   (liquor charge)
---------------------------
     30   (total bill without liquor charge)
*   .15   (tip percentage)
----------------------------
$ 4.50    (tip)
```

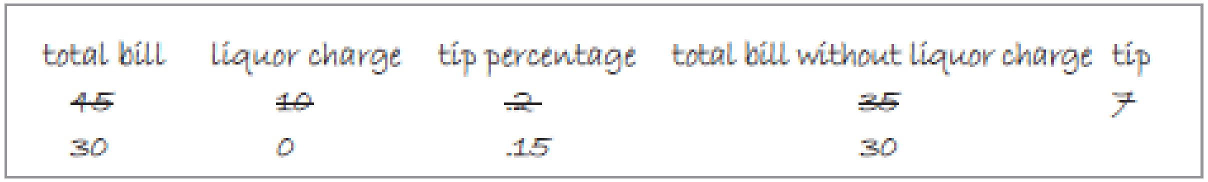
Figure 2-17 Manual tip calculation for the second desk-check

Step 3–Desk-Checking the Algorithm (cont'd.)



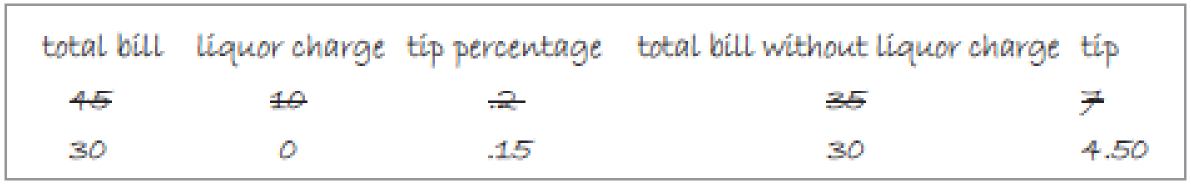
total bill	liquor charge	tip percentage	total bill without liquor charge	tip
~~45~~	~~10~~	~~.2~~	35	7
30	0	.15		

Figure 2-18 Second set of input values entered in the desk-check table



total bill	liquor charge	tip percentage	total bill without liquor charge	tip
~~45~~	~~10~~	~~.2~~	~~35~~	7
30	0	.15	30	

Figure 2-19 Value of the second desk-check's processing item entered in the desk-check table



total bill	liquor charge	tip percentage	total bill without liquor charge	tip
~~45~~	~~10~~	~~.2~~	~~35~~	~~7~~
30	0	.15	30	4.50

Figure 2-20 Value of the second desk-check's output item entered in the desk-check table

Step 3–Desk-Checking the Algorithm (cont'd.)

- **Valid data**: data that the algorithm is expecting the user to enter
- **Invalid data**: data that the algorithm is not expecting the user to enter
- You should test an algorithm with invalid data
  - Users may make mistakes when entering data

The Gas Mileage Problem

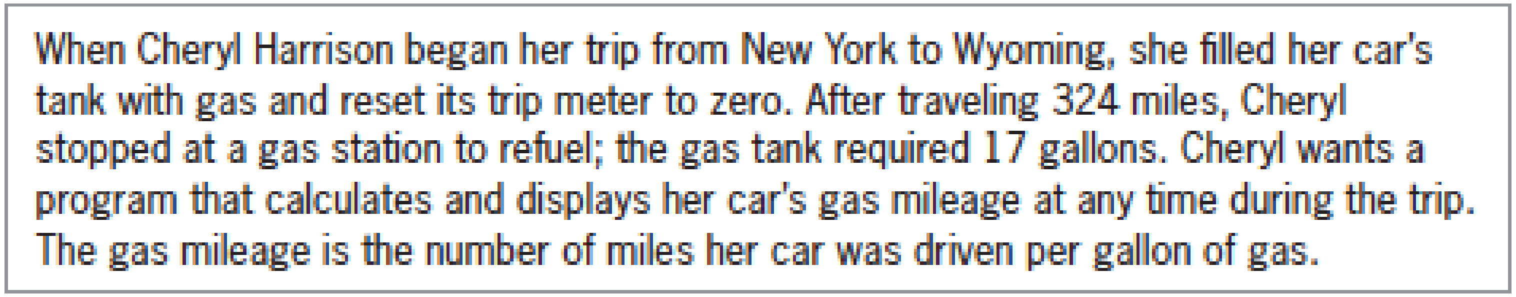
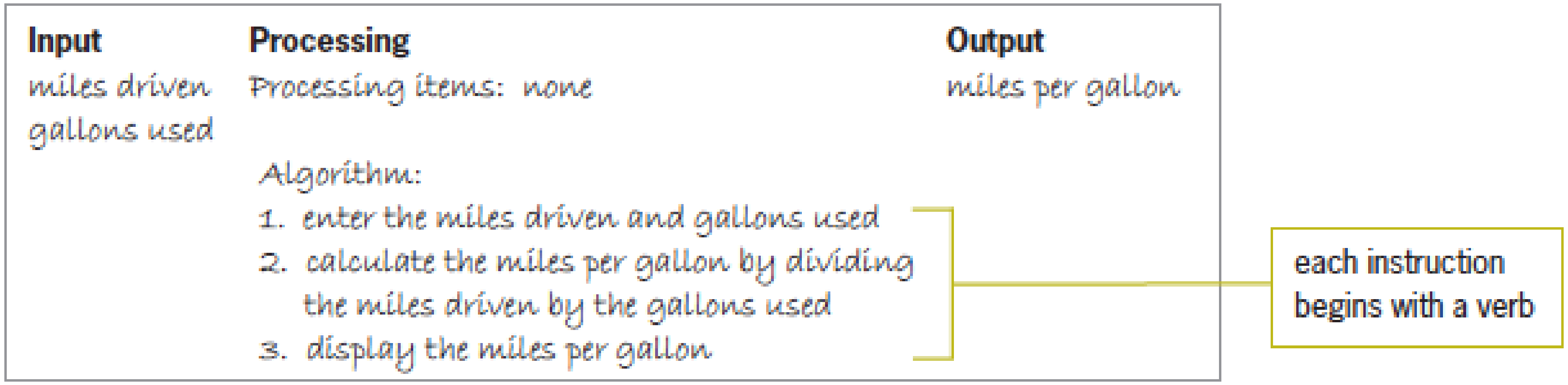

When Cheryl Harrison began her trip from New York to Wyoming, she filled her car's tank with gas and reset its trip meter to zero. After traveling 324 miles, Cheryl stopped at a gas station to refuel; the gas tank required 17 gallons. Cheryl wants a program that calculates and displays her car's gas mileage at any time during the trip. The gas mileage is the number of miles her car was driven per gallon of gas.

Figure 2-21 Problem specification for the gas mileage problem

The Gas Mileage Problem (cont'd.)

- Plan the algorithm with an IPO chart



Input	Processing	Output
miles driven gallons used	Processing items: none	miles per gallon

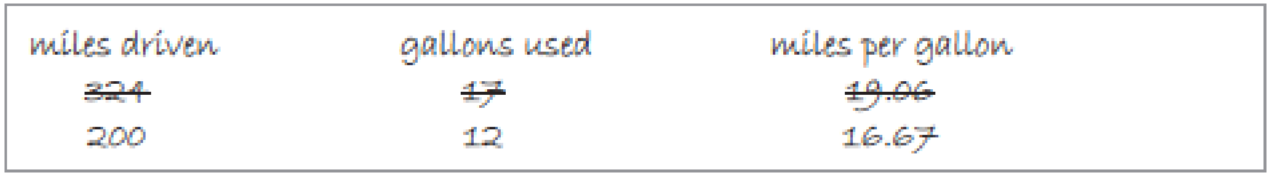
Algorithm:
1. enter the miles driven and gallons used
2. calculate the miles per gallon by dividing the miles driven by the gallons used
3. display the miles per gallon

each instruction begins with a verb

Figure 2-22 IPO chart for the gas mileage problem

The Gas Mileage Problem (cont'd.)

- Then desk-check the algorithm



miles driven	gallons used	miles per gallon
~~324~~	~~17~~	~~19.06~~
200	12	16.67

Figure 2-23 Desk-check table for the gas mileage problem

Summary

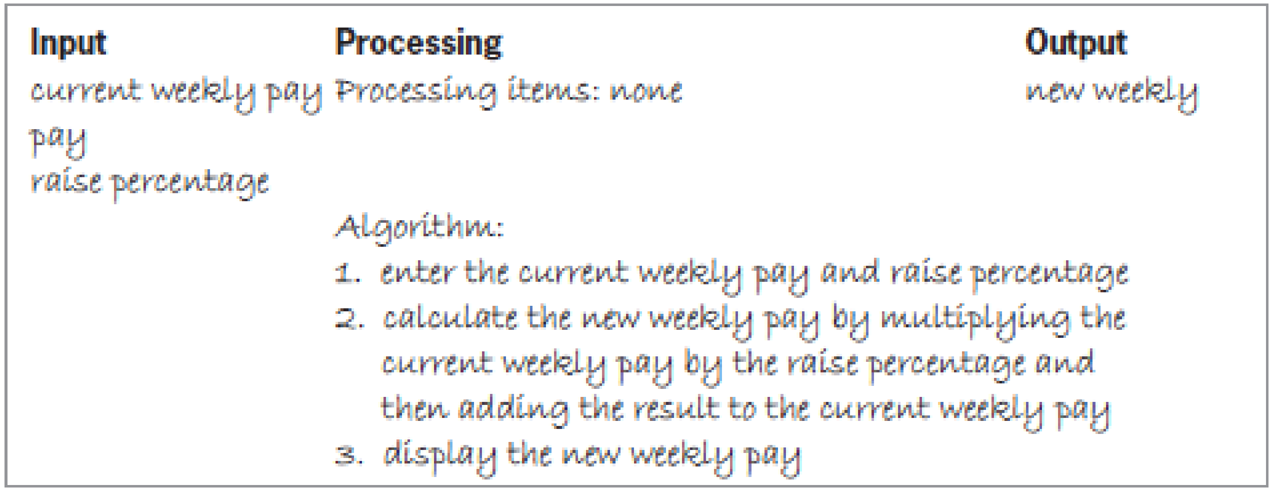
- Problem solving typically involves analyzing the problem and then planning, reviewing, implementing, evaluating, and modifying (if necessary) the solution
- Programmers use tools (IPO charts, pseudocode, flowcharts) to help them analyze problems and develop algorithms
- The first step in problem solving is to analyze the problem
  - First determine the output and then the input

Summary (cont'd.)

- The second step is to plan the algorithm
  - Write the steps that will transform the input into the output
  - Most algorithms begin with entering input data, then processing the data, then displaying the output
- The third step is to desk-check the algorithm
  - Choose sample data and manually compute the expected output
  - Create a desk-check table to fill in values step by step

Lab 2-1: Stop and Analyze

- Aiden Nelinski is paid every Friday and will receive either a 2.0% or 2.5% raise next week
- He wants a program that calculates and displays the amount of his new weekly pay



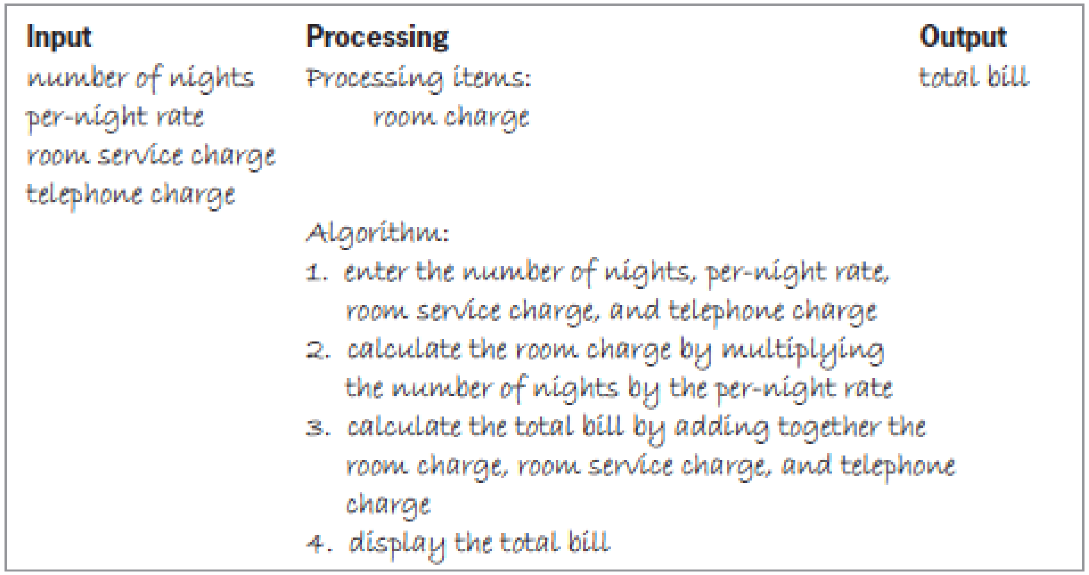
Input	Processing	Output
current weekly pay raise percentage	Processing items: none	new weekly pay

```
Algorithm:
1. enter the current weekly pay and raise percentage
2. calculate the new weekly pay by multiplying the
   current weekly pay by the raise percentage and
   then adding the result to the current weekly pay
3. display the new weekly pay
```

Figure 2-26 IPO chart for Lab 2-1

Lab 2-2: Plan and Create

- Create an algorithm for the manager of the Lakeview Hotel



Input	Processing	Output
number of nights per-night rate room service charge telephone charge	Processing items: room charge Algorithm: 1. enter the number of nights, per-night rate, room service charge, and telephone charge 2. calculate the room charge by multiplying the number of nights by the per-night rate 3. calculate the total bill by adding together the room charge, room service charge, and telephone charge 4. display the total bill	total bill

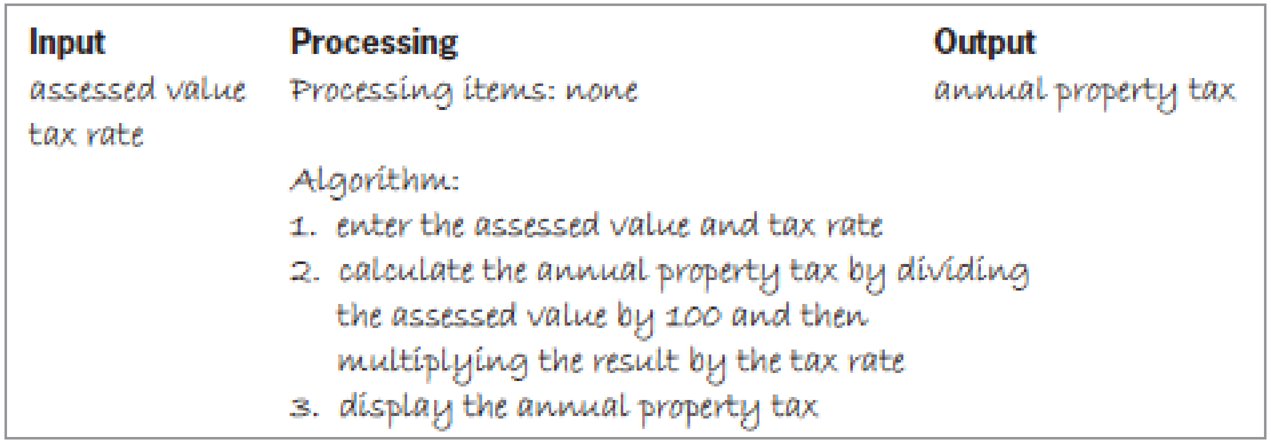
Figure 2-30 Completed IPO chart for Lab 2-2

Lab 2-3: Modify

- Each guest of the Lakeview Hotel pays an entertainment tax, which is a percentage of the room charge only
- Modify the IPO chart in Figure 2-30
- Desk-check the algorithm twice using the given values

Lab 2-4: Desk-Check

- An algorithm is given to calculate and display an annual property tax
- Desk-check the algorithm three times using the given values

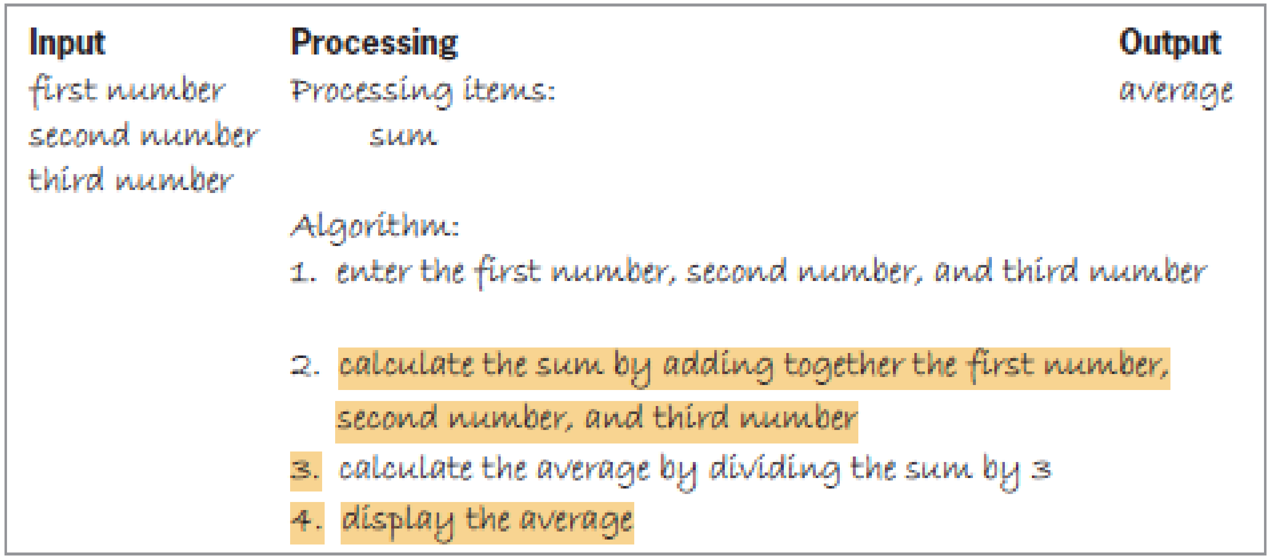


Input	Processing	Output
assessed value tax rate	Processing items: none Algorithm: 1. enter the assessed value and tax rate 2. calculate the annual property tax by dividing the assessed value by 100 and then multiplying the result by the tax rate 3. display the annual property tax	annual property tax

Figure 2-36 IPO chart for Lab 2-4

Lab 2-5: Debug

- An algorithm is given to calculate and display the average of three numbers but is incorrect
- Find and correct the errors in the algorithm



Input
first number
second number
third number

Processing
Processing items:
    sum

Algorithm:
```
1. enter the first number, second number, and third number

2. calculate the sum by adding together the first number,
   second number, and third number
3. calculate the average by dividing the sum by 3
4. display the average
```

Output
average

Figure 2-42 Corrected algorithm for Lab 2-5