An Introduction to Programming with C++
Eighth 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.)

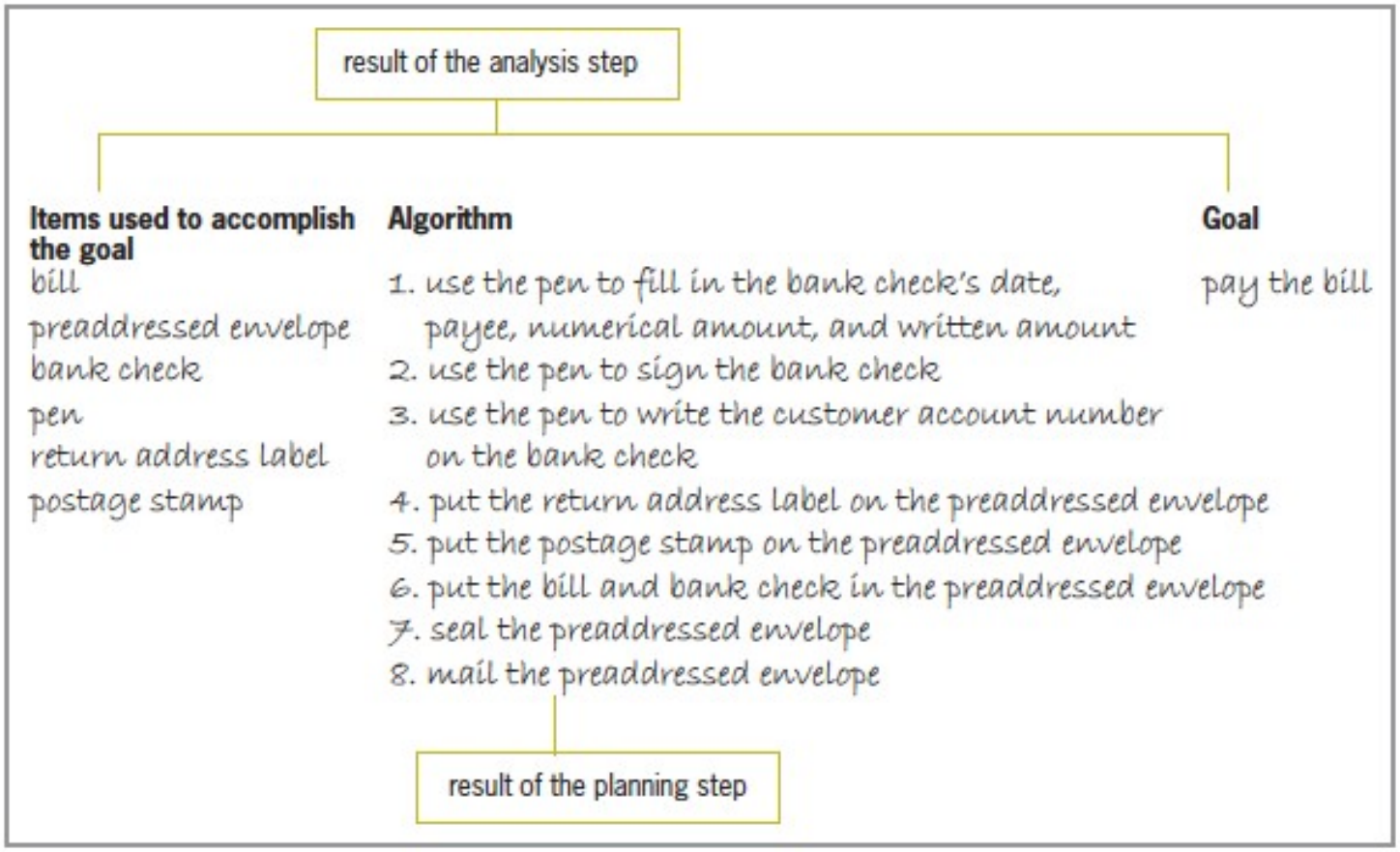


Figure 2-1 Summary of the analysis and planning steps for the bill paying problem

Solving Everyday Problems (cont'd.)

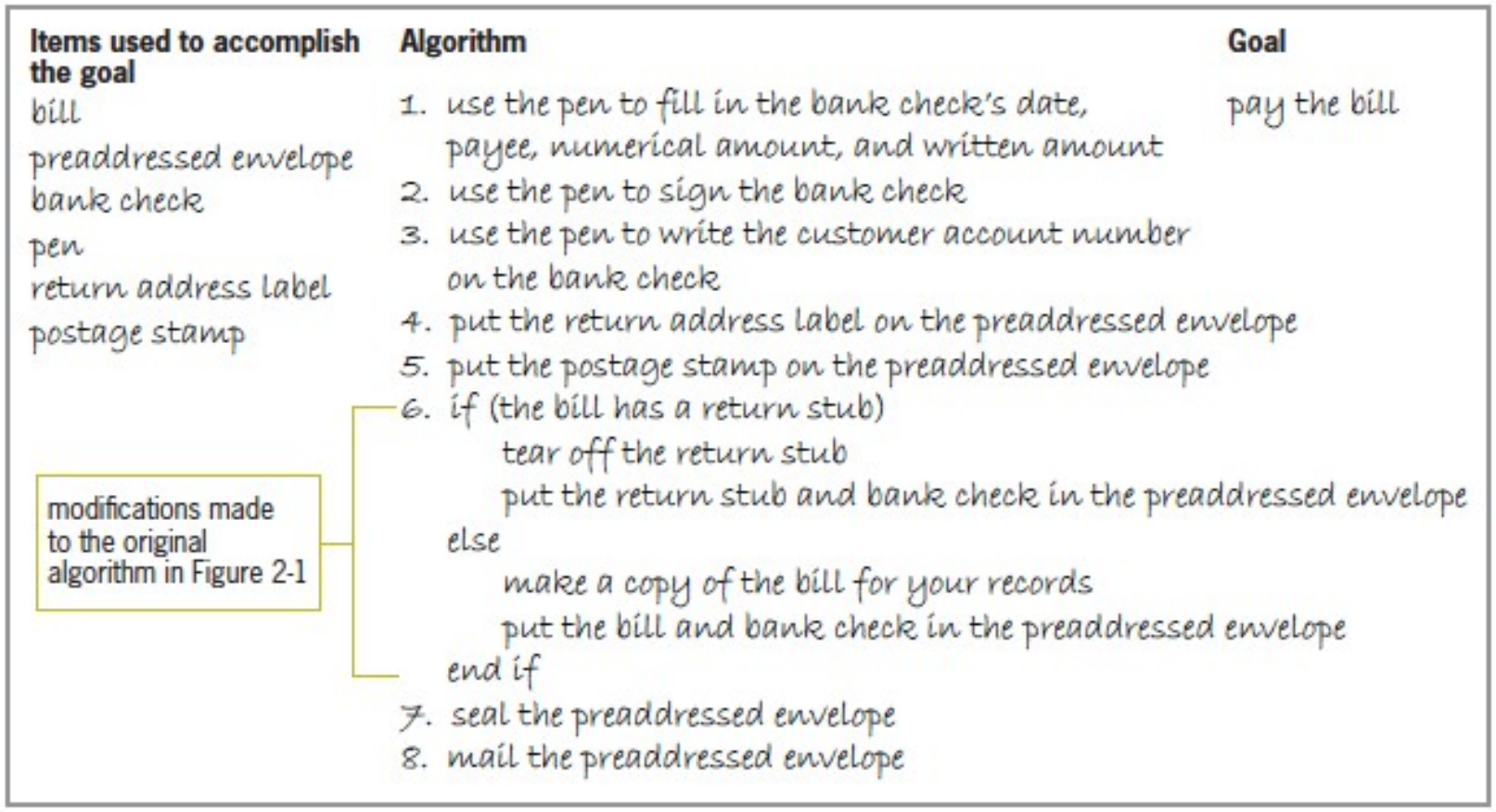


Figure 2-2 Modified algorithm for the bill paying problem

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.)

Addison O'Reilly wants a program that calculates and displays the cost of a 4K Ultra HD TV, which is finally on sale at one of the stores in her area. The program should calculate the cost by multiplying the sale price by the state sales tax rate and then adding the result to the sale price.

Figure 2-4 Problem specification for Addison O'Reilly

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

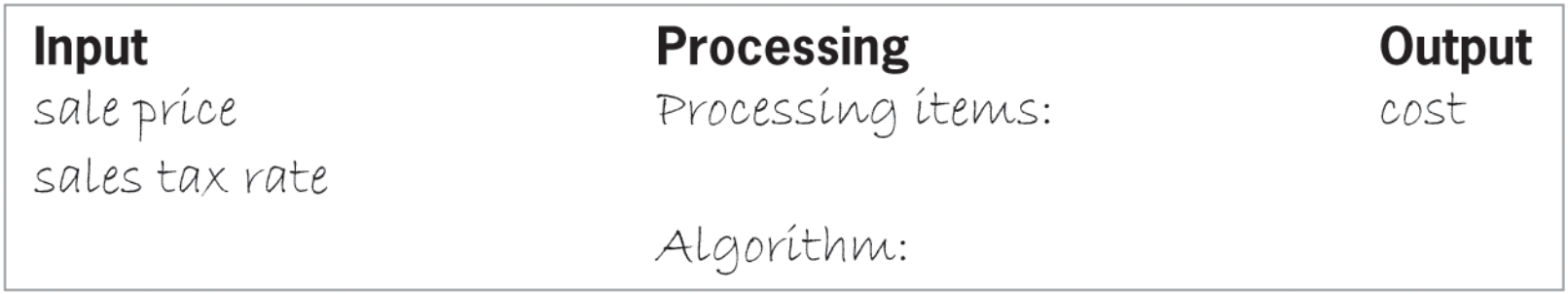


Figure 2-5 Partially completed IPO chart showing the input and output items

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

~~Addison O'Reilly wants a program that~~ calculates and displays the cost ~~of a 4K Ultra HD TV, which is finally on sale at one of the stores in her area. The program should~~ calculate the cost by multiplying the sale price by the state sales tax rate and then adding the result to the sale price.

Figure 2-6 Problem specification with unimportant information crossed out

Hints for Analyzing Problems (cont'd.)

- Some problem specifications contain incomplete information

Cintia Johanson earns $11.50 per hour. Last week, she worked 45 hours. Create a program that calculates and displays her 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

Gordon Matthew 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

Addison O'Reilly wants a program that calculates and displays the cost of a 4K Ultra HD TV, which is finally on sale at one of the stores in her area. The program should calculate the cost by multiplying the sale price by the state sales tax rate and then adding the result to the sale price.

Input	Processing	Output
sale price sales tax rate	Processing items: none	cost

Algorithm:

1. enter the sale price and sales tax rate
2. calculate the cost by multiplying the sale price by the sales tax rate and then adding the result to the sale price
3. display the cost

each instruction begins with a verb

Figure 2-9 Problem specification and IPO chart for the Addison O'Reilly problem

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

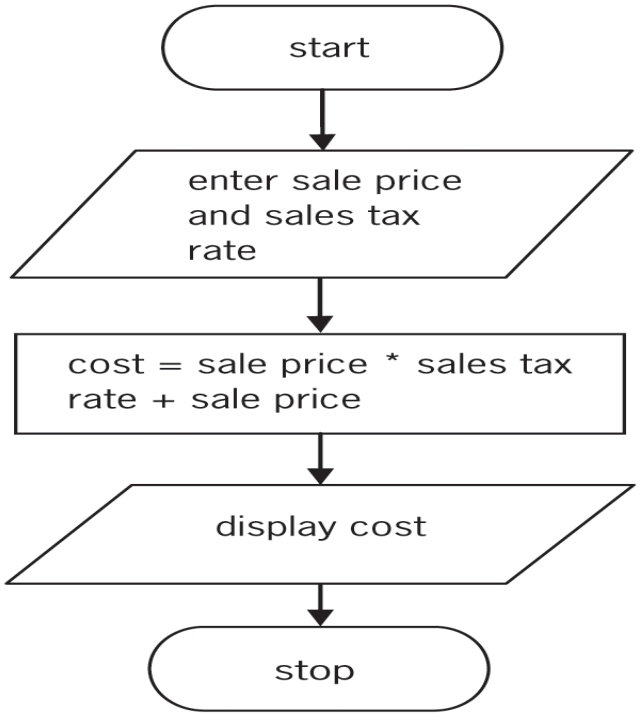
- **Flowcharts** are also used to plan an algorithm
  - Use standardized symbols
  - Symbols connected with **flowlines**
  - Oval: **start/stop symbol**
    - Represents beginning and end of algorithm
  - Rectangle: **process symbol**
    - Represents tasks such as calculations
  - Parallelogram: **input/output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

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

Input
sale price
sales tax rate

Processing
Processing items: none

Algorithm:



Output
cost

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

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

Problem specification

Addison O'Reilly wants a program that calculates and displays the cost of a 4K Ultra HD TV, which is finally on sale at one of the stores in her area. The program should calculate the cost by multiplying the sale price by the state sales tax rate and then adding the result to the sale price.

Input	Processing	Output
sale price sales tax rate	Processing items: sales tax	cost

Algorithm (pseudocode):

1. enter the sale price and sales tax rate
2. calculate the sales tax by multiplying the sale price by the sales tax rate
3. calculate the cost by adding the sales tax to the sale price
4. display the cost

Figure 2-11 A different solution to the Addison O'Reilly 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

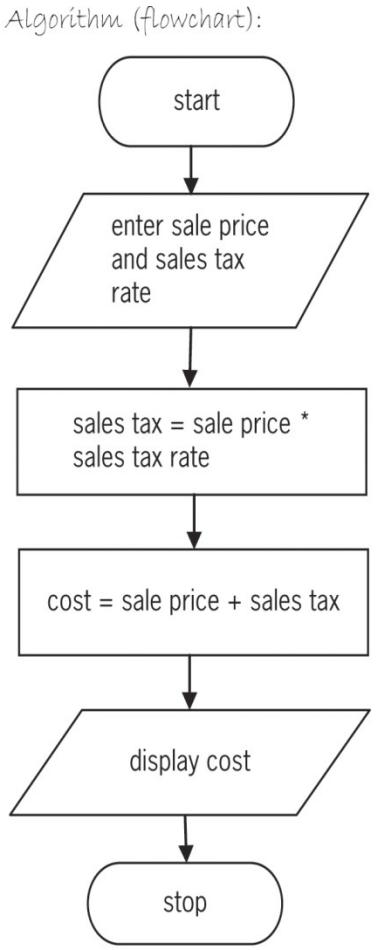


Figure 2-11 A different solution to the Addison O'Reilly problem (flowchart)

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.)

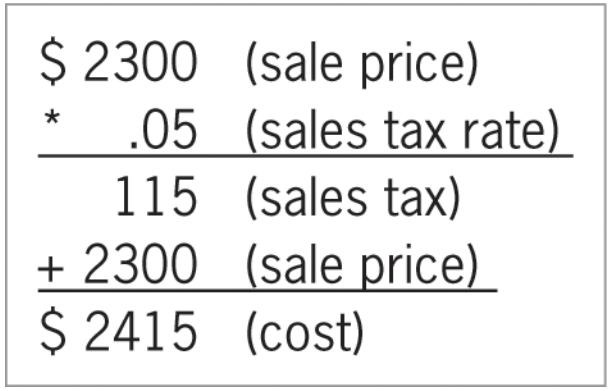
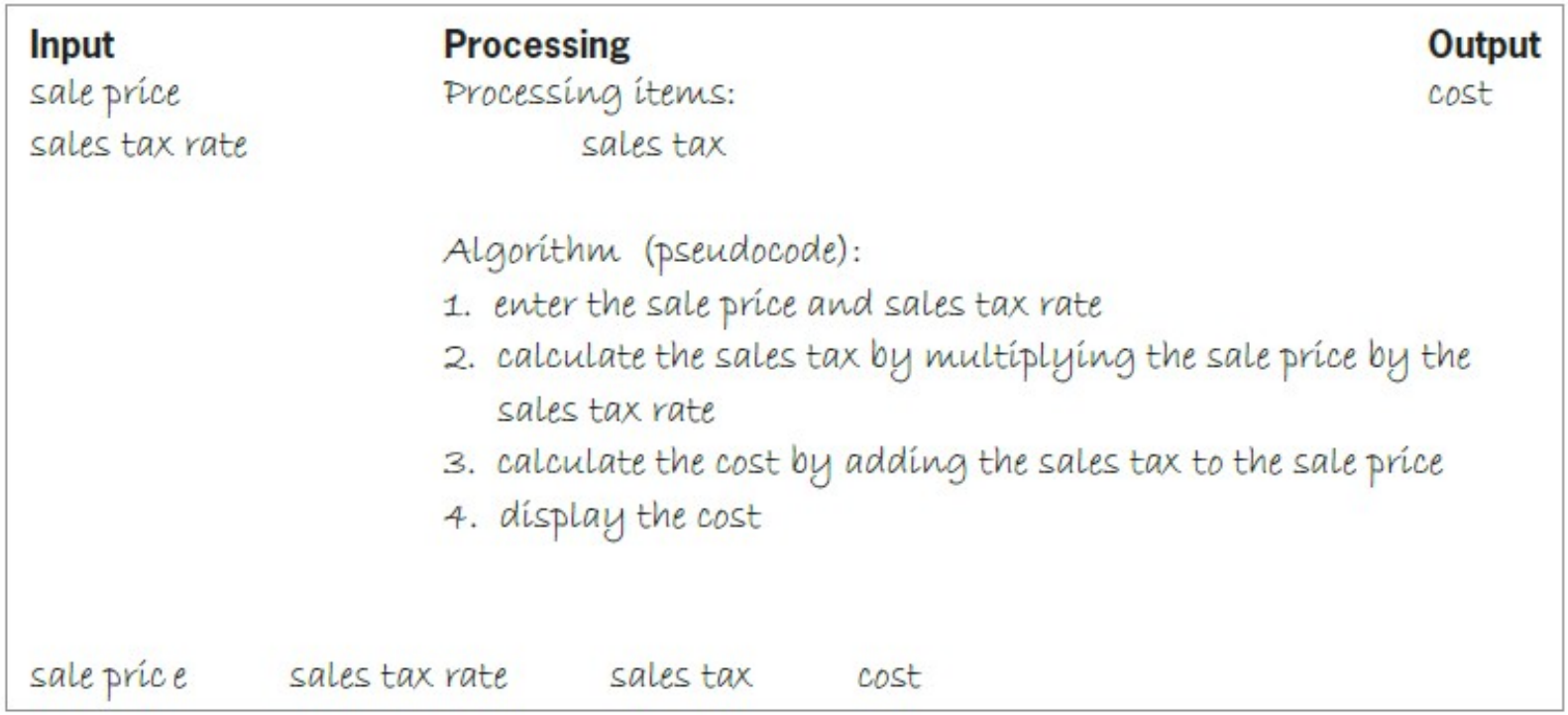


Figure 2-12 Manual cost calculation for the first desk-check

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



Input	Processing	Output
sale price	Processing items:	cost
sales tax rate	sales tax	

Algorithm (pseudocode):

1. enter the sale price and sales tax rate
2. calculate the sales tax by multiplying the sale price by the sales tax rate
3. calculate the cost by adding the sales tax to the sale price
4. display the cost

sale price	sales tax rate	sales tax	cost

Figure 2-13 Addison O'Reilly solution and partially completed desk-check table

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

sale price	sales tax rate	sales tax	cost
2300	.05		

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

sale price	sales tax rate	sales tax	cost
2300	.05	115	2415

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

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

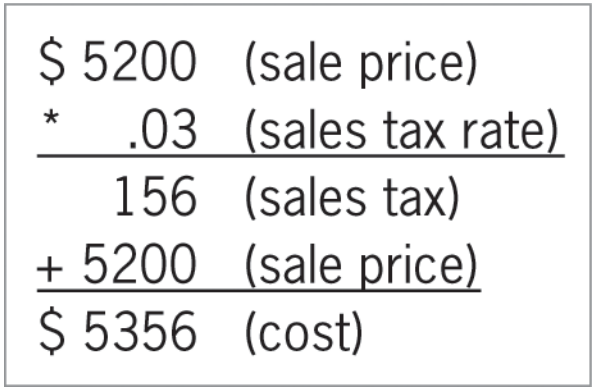


Figure 2-17 Manual cost calculation for the second desk-check

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

sale price	sales tax rate	sales tax	cost
~~2300~~	~~.05~~	115	2415
5200	.03		

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

sale price	sales tax rate	sales tax	cost
~~2300~~	~~.05~~	~~115~~	2415
5200	.03	15	

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

sale price	sales tax rate	sales tax	cost
~~2300~~	~~.05~~	~~115~~	~~2415~~
5200	.03	15	535

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 Sheila Jones began her trip from Vermont to Oregon, she filled her car's tank with gas and reset its trip meter to zero. After traveling 324 miles, Sheila stopped at a gas station to refuel; the gas tank required 17 gallons. Sheila 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 can be 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

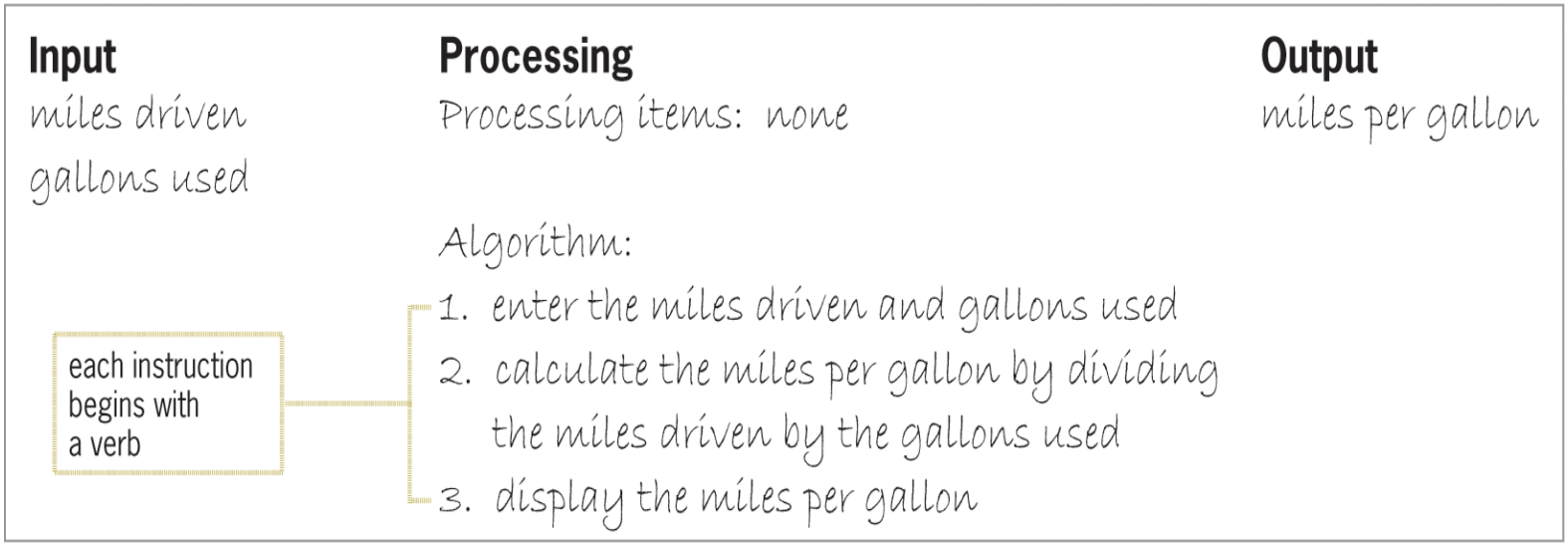


Figure 2-22 IPO chart for the gas mileage problem

The Gas Mileage Problem (cont'd.)

- Then desk-check the algorithm

miles driven	gallons used	miles per gallon
~~324~~	~~17~~	~~19.06~~
400	1	26.67

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

Study the IPO chart shown in Figure 2-26 and then answer the questions.

Input	Processing	Output
quantity sold item cost item selling price	Processing items: price and cost difference Algorithm: 1. enter the quantity sold, item cost, and item selling price 2. calculate the price and cost difference by subtracting the item cost from the item selling price 3. calculate the profit by multiplying the price and cost difference by the quantity sold 4. display the profit	profit

quantity sold item cost item selling price price and cost difference profit

Figure 2-26 IPO chart for Lab 2-1

Lab 2-1: Stop and Analyze (cont'd.)

- Complete the desk-check table using two sets of input values. First, use 100, 5, and 8 as the quantity sold, item cost, and item selling price, respectively. The use 650, 2.50, and 3.75. What will the algorithm display using the first set of input values? What will the algorithm display using the second set of input values?
- How would you modify the IPO chart to also display the difference between the item cost and item selling price?
- How would you modify the IPO chart and desk-check table to eliminate the use of a processing item?

Lab 2-2: Plan and Create

- Create an algorithm for the manager of the Jericho Bakery

Jericho Bakery sells a variety of doughnuts and muffins for $0.50 each. The manager of the bakery wants a program that calculates and displays the total number of items (doughnuts and muffins) purchased by a customer, as well as the total cost of the order.

Figure 2-27 Problem Specification for Lab 2-2

Lab 2-2: Plan and Create (cont'd.)

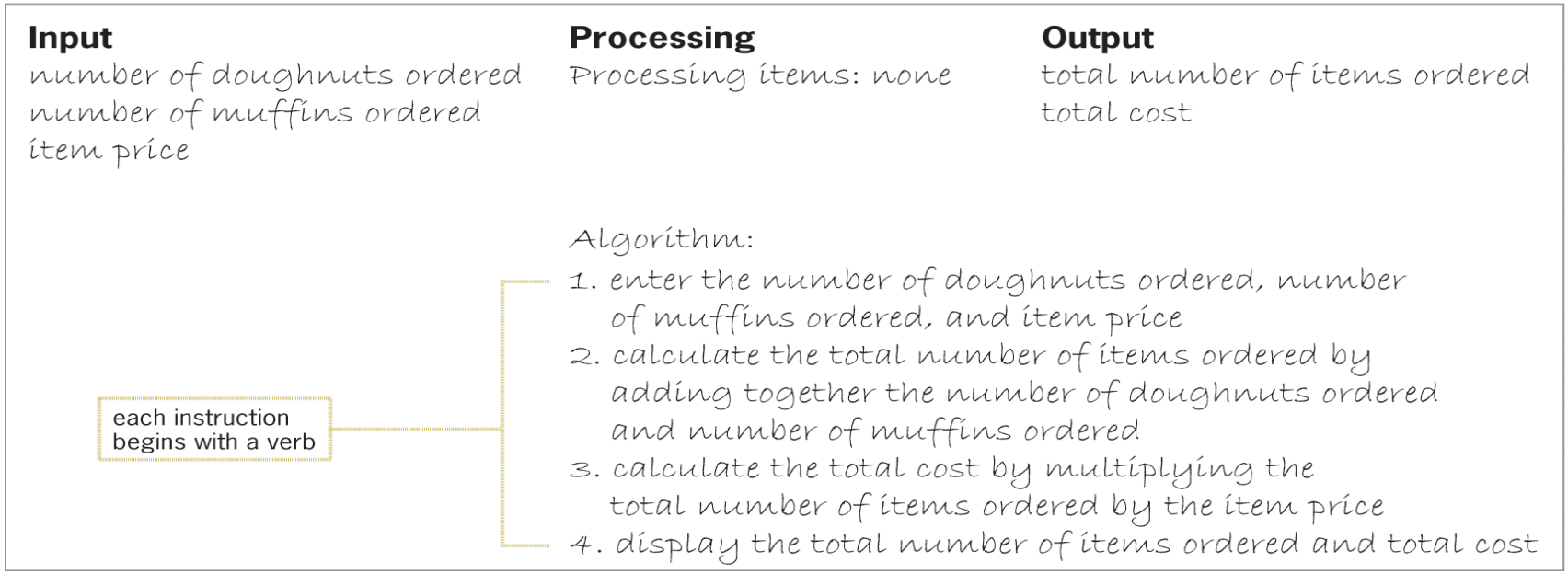


Figure 2-29 Completed IPO Chart for Lab 2-2

Lab 2-2: Plan and Create (cont'd.)

First desk-check

```
    4   (number of doughnuts ordered)
+   2   (number of muffins ordered)
-------
    6   (total number of items ordered)
* 0.50  (item price)
-------
$ 3.00  (total cost)
```

Second desk-check

```
    0   (number of doughnuts ordered)
+  12   (number of muffins ordered)
-------
   12   (total number of items ordered)
* 0.45  (item price)
-------
$ 5.40  (total cost)
```

Figure 2-30 Manual calculations for the two desk-checks

Lab 2-3: Modify

- Jericho Bakery has raised the price of its muffins from $0.50 to $0.55.
- Make the appropriate modifications to the IPO chart shown earlier in Figure 2-29.
- Desk-check the algorithm twice.
- For the first desk-check, use 4, 2, 0.50, and 0.55 as the number of doughnuts ordered, number of muffins ordered, doughnut price, and muffin price, respectively.
- For the second desk-check, use 0, 12, 0.60, and 0.70.

Lab 2-4: What's Missing?

A local club sells boxes of three types of cookies: shortbread, pecan sandies, and chocolate mint. The club leader wants a program that displays the percentage that each of the cookie types contributes to the total cookie sales. Figure 2-34 contains a list of items and instructions that you can use for this lab. Fill in the IPO chart and determine whether any items are missing from the list. Next, put the instructions in the proper order and determine the one or more missing instructions.

Lab 2-4: What's Missing? (cont'd.)

Input	Processing	Output
	Processing items: Algorithm:	

Items

chocolate mint contribution
chocolate mint sold
pecan sandies contribution
pecan sandies sold
shortbread contribution
shortbread sold

Instructions

calculate chocolate mint contribution by dividing chocolate mint sold by total sold, and then multiplying the result by 100
calculate pecan sandies contribution by dividing pecan sandies sold by total sold, and then multiplying the result by 100
calculate shortbread contribution by dividing shortbread sold by total sold, and then multiplying the result by 100
display shortbread contribution, pecan sandies contribution, and chocolate mint contribution
enter shortbread sold, pecan sandies sold, and chocolate mint sold

Lab 2-5: Desk-Check

The algorithm in Figure 2-35 displays three suggested amounts to tip a waiter. Desk-check the algorithm twice. First, use $102.50 and $5.80 as the bill amount and sales tax, respectively. Then, use $56.78 and $2.18.

Input	Processing	Output
restaurant bill	Processing items:	10% tip
sales tax	bill before sales tax	15% tip
		20% tip

Algorithm:
1. enter the restaurant bill
2. calculate the bill before sales tax by subtracting the sales tax from the restaurant bill
3. calculate the 10% tip by multiplying the bill before sales tax by 10%
4. calculate the 15% tip by multiplying the bill before sales tax by 15%
5. calculate the 20% tip by multiplying the bill before sales tax by 20%
6. display the 10% tip, 15% tip, and 20% tip

Figure 2-35 IPO chart for Lab 2-5

Lab 2-6: Debug

The algorithm below should calculate and display the average of three numbers, but it is not working properly. In this lab, you will find and correct the errors in the algorithm.

Input	Processing	Output
first number second number third number	Processing items: sum	average

Algorithm:

1. enter the first number, second number, and third number
2. calculate the average by dividing the sum by 3
3. display the average number

Figure 2-36 IPO chart for Lab 2-6