

Chapter 4). Finally, it should display the contents of each structure. A sample run should look something like the following:

```
How many cars do you wish to catalog? 2
Car #1:
Please enter the make: Hudson Hornet
Please enter the year made: 1952
Car #2:
Please enter the make: Kaiser
Please enter the year made: 1951
Here is your collection:
1952 Hudson Hornet
1951 Kaiser
```

8. Write a program that uses an array of `char` and a loop to read one word at a time until the word `done` is entered. The program should then report the number of words entered (not counting `done`). A sample run could look like this:

```
Enter words (to stop, type the word done):
anteater birthday category dumpster
envy finagle geometry done for sure
You entered a total of 7 words.
```

You should include the `cstring` header file and use the `strcmp()` function to make the comparison test.

9. Write a program that matches the description of the program in Programming Exercise 8, but use a `string` class object instead of an array. Include the `string` header file and use a relational operator to make the comparison test.
10. Write a program using nested loops that asks the user to enter a value for the number of rows to display. It should then display that many rows of asterisks, with one asterisk in the first row, two in the second row, and so on. For each row, the asterisks are preceded by the number of periods needed to make all the rows display a total number of characters equal to the number of rows. A sample run would look like this:

```
Enter number of rows: 5
.....*
...**
..***
.****
*****
```

Branching Statements and Logical Operators

In this chapter you'll learn about the following:

- The `if` statement
- The `if else` statement
- Logical operators: `&&`, `||`, and `!`
- The `cctype` library of character functions
- The conditional operator: `?:`
- The `switch` statement
- The `continue` and `break` statements
- Number-reading loops
- Basic file input/output

One of the keys to designing intelligent programs is to give them the ability to make decisions. Chapter 5, “Loops and Relational Expressions,” shows one kind of decision making—looping—in which a program decides whether to continue looping. This chapter investigates how C++ lets you use branching statements to decide among alternative actions. Which vampire-protection scheme (garlic or cross) should the program use? What menu choice has the user selected? Did the user enter a zero? C++ provides the `if` and `switch` statements to implement decisions, and they are this chapter's main topics. This chapter also looks at the conditional operator, which provides another way to make a choice, and the logical operators, which let you combine two tests into one. Finally, the chapter takes a first look at file input/output.