

# CHAPTER 6

## NESTED SELECTION STRUCTURES

The answers for the Nested Selection Structures section are located at the end of the section.

1. WM-Figure 6-1 shows an algorithm involving a robot named Rob. The algorithm shows Rob how to say the words "hello" and "good bye" in Spanish. Modify the algorithm so that it determines whether Rob wants to say the words in either Spanish or French. The French words for "hello" and "good bye" are "bonjour" and "au revoir," respectively.

```
if (you want to say "hello")
    say "hola"
else
    say "adios"
end if
```

**WM-Figure 6-1** Algorithm for the Rob problem

2. WM-Figure 6-2 shows an algorithm for the Haperin Company. Modify the algorithm so that it calculates a 5% commission when the sales are at least \$25000, but a 6% commission when the sales are over \$60000. If the sales are less than \$25000, the commission should be 0.

```
Algorithm:
1. enter the sales
2. calculate the commission by multiplying the sales by .05
3. display the commission
```

**WM-Figure 6-2** Solution for the Haperin Company

3. WM-Figure 6-3 shows an algorithm for Budget Sales. Modify the algorithm so that it calculates a 7% commission when the sales are at least \$40000. If the sales are less than \$40000, calculate a 1% commission if the salesperson has been with the company for over 3 years; otherwise, the commission should be 0.

```
Algorithm:
1. enter the sales and years employed
2. calculate the commission by multiplying the sales by .07
3. display the commission
```

**WM-Figure 6-3** Solution for Budget Sales

- Analyze the problem specification shown in WM-Figure 6-4.

Your friend, Jacob, wants a program that calculates and displays the balance in his checking account, given the beginning balance, the amount of money he deposited, and the amount of money he withdrew. If the ending balance is less than 0, the program should also display one of two different messages. The appropriate message depends on whether Jacob has overdraft protection at his bank. If he has the protection, display the “Bank will cover” message; otherwise, display the “Get to the bank” message.

**WM-Figure 6-4** Problem specification for Jacob

- Analyze the problem specification shown in WM-Figure 6-5.

Professor Jerome wants a program that displays various messages. The messages depend on the test score entered by the professor. If the test score is at least 70, display the “Pass” message. If the test score is at least 60, but less than 70, display the “Retake test” message. If the test score is below 60, display the “Fail” message.

**WM-Figure 6-5** Problem specification for Professor Jerome

- Rewrite the algorithm you created for Professor Jerome in a different way.
- Analyze the problem specification shown in WM-Figure 6-6.

South Carlton Bank wants a program that displays the bank’s service fee for a checking account. The basic fee is \$2 per month; however, senior citizens do not pay the basic fee. The bank also charges \$.02 per check for the first 6 checks written, and then \$.10 for each additional check. Senior citizens, however, pay \$.01 for each check. The user will enter the number of checks written and whether the customer is a senior citizen.

**WM-Figure 6-6** Problem specification for South Carlton Bank

- Analyze the problem specification shown in WM-Figure 6-7.

Yonger College wants a program that displays either the “Accepted” or “Not Accepted” message. The appropriate message depends on the student’s SAT score and GPA, which are entered by the user. The “Accepted” message should be displayed when the SAT score is at least 1500. When the SAT score is less than 1500, but at least 1350, display the “Accepted” message if the GPA is at least 3.5. Display the “Not Accepted” message in all other cases.

**WM-Figure 6-7** Problem specification for Yonger College

- Analyze the problem specification shown in WM-Figure 6-8.

Universal Surveys wants a program that displays the number of males and females responding to a survey, as well as the number of respondents who didn’t provide their gender. The user will enter 1 for Male, 2 for Female, or 3 for Unanswered.

**WM-Figure 6-8** Problem specification for Universal Surveys

10. Park and Company pays a monthly bonus (either \$50 or \$150) to its salespeople. The bonus is based on the number of items the salesperson sold. To receive \$50, the salesperson must sell at least 100 items. To receive \$150, the salesperson must sell at least 2000 items. Compare the four algorithms shown in WM-Figure 6-9. Which algorithm(s) will display the correct bonus amount? If an algorithm will not work correctly, explain why it is incorrect.

<p>Algorithm 1:</p> <ol style="list-style-type: none"><li>1. enter the number sold</li><li>2. if (number sold is at least 100)     assign 50 as the bonus     else         if (number sold is at least 2000)             assign 150 as the bonus         else             assign 0 as the bonus         end if     end if 3. display the bonus</li></ol>	<p>Algorithm 2:</p> <ol style="list-style-type: none"><li>1. enter the number sold</li><li>2. if (number sold is less than 100)     assign 0 as the bonus     else         if (number sold is at least 2000)             assign 150 as the bonus         else             assign 50 as the bonus         end if     end if 3. display the bonus</li></ol>
<p>Algorithm 3:</p> <ol style="list-style-type: none"><li>1. enter the number sold</li><li>2. if (number sold is less than 100)     assign 0 as the bonus     else         if (number sold is at least 100)             assign 50 as the bonus         else             assign 150 as the bonus         end if     end if 3. display the bonus</li></ol>	<p>Algorithm 4:</p> <ol style="list-style-type: none"><li>1. enter the number sold</li><li>2. if (number sold is at least 2000)     assign 150 as the bonus     else         if (number sold is at least 100)             assign 50 as the bonus         else             assign 0 as the bonus         end if     end if 3. display the bonus</li></ol>

**WM-Figure 6-9** Algorithms for Park and Company

## ANSWERS FOR THE PROBLEM SPECIFICATIONS SECTION

1.

```
if (you want to speak Spanish)
    if (you want to say "hello")
        say "hola"
    else
        say "adios"
    end if
else
    if (you want to say "hello")
        say "bonjour"
    else
        say "au revoir"
    end if
end if
```

**WM-Figure 6-10** Modified algorithm for the Rob problem

2.

```
1. enter the sales
2. if (the sales are over 60000)
    calculate the commission by multiplying the sales by .06
else
    if (the sales are at least 25000)
        calculate the commission by multiplying the sales by .05
    else
        assign 0 as the commission
    end if
end if
3. display the commission
```

**WM-Figure 6-11** Modified algorithm for the Haperin Company

3.

```
1. enter the sales and years employed
2. if (the sales are at least 40000)
    calculate the commission by multiplying the sales by .07
else
    if (the years employed are over 3)
        calculate the commission by multiplying the sales by .01
    else
        assign 0 as the commission
    end if
end if
3. display the commission
```

**WM-Figure 6-12** Modified algorithm for Budget Sales

4.

Input	Processing	Output
beginning balance amount deposited amount withdrawn overdraft protection status (Y or N)	Processing items: none  Algorithm: 1. enter the beginning balance, amount deposited, and amount withdrawn 2. calculate the ending balance by adding the amount deposited to the beginning balance, and then subtracting the amount withdrawn from the result 3. display the ending balance 4. if (the ending balance is less than 0) if (overdraft protection is Y) display "Bank will cover" message else display "Get to the bank" message end if end if	ending balance message ("Bank will cover" or "Get to the bank")

WM-Figure 6-13 Solution for the Jacob problem

5.

Input	Processing	Output
test score	Processing items: none  Algorithm: 1. enter test score 2. if (the test score is at least 70) display "Pass" else if (the test score is at least 60) display "Retake test" else display "Fail" end if end if	message ("Pass", "Retake test", of "Fail")

WM-Figure 6-14 Solution for the Professor Jerome problem

6.

```

Algorithm:
1. enter test score
2. if (the test score is less than 60)
    display "Fail"
else
    if (the test score is at least 70)
        display "Pass"
    else
        display "Retake test"
    end if
end if
  
```

**WM-Figure 6-15** Different algorithm for the Professor Jerome problem

7.

Input	Processing	Output
number of checks written senior status (Y or N)	Processing items: check charge basic fee  Algorithm: 1. enter the number of checks written and senior status 2. if (senior status is Y) calculate the check charge by multiplying the number of checks written by .01 assign 0 as the basic fee else assign 2 as the basic fee if (the number of checks written is less than or equal to 6) calculate the check charge by multiplying the number of checks written by .02 else calculate the check charge by subtracting 6 from the number of checks written, then multiplying the result by .10, and then adding .12 to the result end if end if 3. calculate the service fee by adding the basic fee to the check charge 4. display the service fee	service fee

**WM-Figure 6-16** Solution for the South Carlton Bank problem

8.

Input	Processing	Output
SAT score GPA	Processing items: none  Algorithm: 1. enter the SAT score and GPA 2. if (SAT score is at least 1500) display "Accepted" else if (SAT score is at least 1350) if (GPA is at least 3.5) display "Accepted" end if else display "Not Accepted" end if end if	message (either "Accepted" or "Not Accepted")

WM-Figure 6-17 Solution for the Yonger College problem

9.

Input	Processing	Output
gender (1 for male, 2 for female, 3 for unanswered)	Processing items: none  Algorithm: 1. enter the gender 2. if (gender is 1) add 1 to the number of males else if (gender is 2) add 1 to the number of females else add 1 to the number of unanswered end if end if 3. display the number of males, number of females, and number of unanswered	number of males number of females number of unanswered

WM-Figure 6-18 Solution for the Universal Surveys problem

10. Algorithms 2 and 4 will display the correct bonus amount. Algorithms 1 and 3 will not work correctly because both will assign the \$50 bonus (rather than the \$150 bonus) to salespeople who sell at least 2000.

## MULTIPLE-ALTERNATIVE SELECTION STRUCTURES

The answers for the Multiple-Alternative Selection Structures section are located at the end of the section.

1. Write the code for a multiple-alternative selection structure that displays a name based on the employee code entered by the user. The employee code is stored in an `int` variable named `empCode`. The employee codes and names are show here. If the employee code does not appear in the list, the selection structure should display the "Invalid code" message. Use the shorter form of the `if` statement.

<u>Code</u>	<u>Name</u>
1	Sam
2	Jack
3	Sue
4	Sue
5	Mary

2. Rewrite the previous selection structure using the `switch` statement.
3. Write the code for a multiple-alternative selection structure that determines an employee's gross pay for each pay period. Code 1 employees are paid monthly, which means they receive 12 paychecks. Code 2 employees are paid twice per month, so they receive 24 paychecks. Code 3 employees are paid weekly and receive 52 paychecks. The employee's annual salary and code are stored in the `salary` and `empCode` variables. The `salary` variable has the `double` data type. The `empCode` variable has the `char` data type. Assign the gross pay to a `double` variable named `gross`. If the employee code is not 1, 2 or 3, the selection structure should assign 0 as the gross pay. Use the shorter form of the `if` statement.
4. Rewrite the previous selection structure using the `switch` statement.

## ANSWERS FOR THE MULTIPLE-ALTERNATIVE SELECTION STRUCTURES SECTION

1. 

```
if (empCode == 1)
    cout << "Sam";
else if (empCode == 2)
    cout << "Jack";
else if (empCode == 3 || empCode == 4)
    cout << "Sue";
else if (empCode == 5)
    cout << "Mary";
else
    cout << "Invalid code";
//end if
```



```

2.  switch (empCode)
    {
    case 1:
        cout << "Sam";
        break;
    case 2:
        cout << "Jack";
        break;
    case 3:
    case 4:
        cout << "Sue";
        break;
    case 5:
        cout << "Mary";
        break;
    default:
        cout << "Invalid code";
    } //end switch

3.  if (empCode == '1')
        gross = salary / 12;
    else if (empCode == '2')
        gross = salary / 24;
    else if (empCode == '3')
        gross = salary / 52;
    else
        gross = 0;
    // end if

4.  switch (empCode)
    {
    case '1':
        gross = salary / 12;
    case '2':
        gross = salary / 24;
    case '3':
        gross = salary / 52;
    default:
        gross = 0;
    } // end switch

```