Answers to Chapter 1 Mini-Quizzes

**Mini-Quiz 1-1**

1. machine
2. a. procedure-oriented
3. b. object-oriented
4. compiler

**Mini-Quiz 1-2**

1. sequence, selection, repetition
2. sequence
3. algorithm
4. repetition
5. repetition
6. selection
Answers to Chapter 1 Labs

**LAB 1-1  Stop and Analyze**

1. sequence and repetition
2. 2100, 315, and 1785
3. Change the first line to repeat for (the first 10 customers buying a TV).
4. Change the second line to enter the original price of the TV and the discount rate. Then, replace 15% in the third line with the discount rate.

**LAB 1-2  Plan and Create**

repeat (10 times)
    enter the salesperson’s name and sales
    if (the sales are greater than 10,000)
        calculate the bonus by multiplying the sales by 10%
    else
        calculate the bonus by multiplying the sales by 5%
    end if
    display the salesperson’s name and bonus
end repeat

**LAB 1-3  Modify**

You can use either of the following algorithms. The modifications are shaded in each.

**Algorithm 1**

repeat for (each customer buying a TV)
    enter the original price of the TV
    if (the customer is a store employee)
        calculate the discount by multiplying the original price by 25%
    else
        calculate the discount by multiplying the original price by 15%
    end if
    calculate the total due by subtracting the discount from the original price
    print a bill showing the original price, discount, and total due
end repeat
Algorithm 2

repeat for (each customer buying a TV)
  enter the original price of the TV
  if (the customer is not a store employee)
    calculate the discount by multiplying the original price by 15%
  else
    calculate the discount by multiplying the original price by 25%
  end if
  calculate the total due by subtracting the discount from the original price
  print a bill showing the original price, discount, and total due
end repeat

LAB 1-4 What’s Missing?
The missing instruction is shaded.

repeat (3 times)
  walk forward one complete step
end repeat
if (Ginger is on the bench)
  gently shove Ginger off the bench
end if
repeat (2 times)
  turn left 90 degrees
end repeat
sit down on the bench

Answers to Chapter 2 Mini-Quizzes

Mini-Quiz 2-1

1. Output: raise
   Input: new salary
   current salary
   raise percentage

2. Output: average score
   Input: midterm score
   final score

3. Output: 10% tip
   15% tip
   20% tip
   Input: customer bill

4. Output: yearly savings
   Input: amount saved per day
   number of days in the year
Mini-Quiz 2-2

1. input/output
2. rectangular
3.

**Input**
- current salary
- raise percentage

**Processing**
- Processing items: none

**Output**
- raise
- new salary

Algorithm:
1. enter the current salary and raise percentage
2. calculate the raise by multiplying the current salary by the raise percentage
3. calculate the new salary by adding the raise to the current salary
4. display the raise and new salary

4.

**Input**
- midterm score
- final score

**Processing**
- Processing items:
  - sum

**Output**
- average score

Algorithm:
```
start
  enter midterm score and final score
  sum = midterm score + final score
  average score = sum / 2
  display average score
stop
```
### Mini-Quiz 2-3

1. 

<table>
<thead>
<tr>
<th>current salary</th>
<th>raise percentage</th>
<th>raise</th>
<th>new salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>$54,700</td>
<td>.02</td>
<td>$104</td>
<td>$55,794</td>
</tr>
</tbody>
</table>

2. 

<table>
<thead>
<tr>
<th>midterm score</th>
<th>final score</th>
<th>sum</th>
<th>average score</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>83</td>
<td>158</td>
<td>79</td>
</tr>
<tr>
<td>98</td>
<td>93</td>
<td>191</td>
<td>95.5</td>
</tr>
</tbody>
</table>

### Answers to Chapter 2 Labs

#### LAB 2-1 Stop and Analyze

1. 

<table>
<thead>
<tr>
<th>quantity sold</th>
<th>item cost</th>
<th>item selling price</th>
<th>price and cost difference</th>
<th>profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>6.50</td>
<td>8.75</td>
<td>2.25</td>
<td>300.00</td>
</tr>
</tbody>
</table>

The algorithm will display 300 using the first set of input values. It will display 812.50 using the second set of input values.

2. 

**Input**
- quantity sold
- item cost
- item selling price

**Processing**
- Processing items: none

**Output**
- price and cost difference
- profit

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 price and cost difference and profit

3. 

**Input**
- quantity sold
- item cost
- item selling price

**Processing**
- Processing items: none

**Output**
- profit

Algorithm:
1. enter the quantity sold, item cost, and item selling price
2. calculate the profit by subtracting the item cost from the item selling price, and then multiplying the result by the quantity sold
3. display the profit

<table>
<thead>
<tr>
<th>quantity sold</th>
<th>item cost</th>
<th>item selling price</th>
<th>profit</th>
</tr>
</thead>
</table>
LAB 2-2 Plan and Create

No answer required.

LAB 2-3 Modify

**Input**
- number of doughnuts ordered
- number of muffins ordered
- doughnut price
- muffin price

**Processing**
Processing items:
- doughnut cost
- muffin cost

**Output**
- total number of items ordered
- total cost

Algorithm:
1. enter the number of doughnuts ordered, number of muffins ordered, doughnut price, and muffin price
2. calculate the total number of items ordered by adding together the number of doughnuts ordered and number of muffins ordered
3. calculate the doughnut cost by multiplying the number of doughnuts ordered by the doughnut price
4. calculate the muffin cost by multiplying the number of muffins ordered by the muffin price
5. calculate the total cost by adding the doughnut cost to the muffin cost
6. display the total number of items ordered and total cost

<table>
<thead>
<tr>
<th>number of doughnuts ordered</th>
<th>number of muffins ordered</th>
<th>doughnut price</th>
<th>muffin price</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>12</td>
<td>0.50</td>
<td>0.55</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.60</td>
<td>0.70</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>doughnut cost</th>
<th>muffin cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8.40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>total number of items ordered</th>
<th>total cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>8.40</td>
</tr>
</tbody>
</table>
### LAB 2-4  What's Missing?

<table>
<thead>
<tr>
<th>Input</th>
<th>Processing</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>shortbread sold</td>
<td>Processing items:</td>
<td>shortbread contribution</td>
</tr>
<tr>
<td>pecan sandies sold</td>
<td>total sold</td>
<td>pecan sandies contribution</td>
</tr>
<tr>
<td>chocolate mint sold</td>
<td></td>
<td>chocolate mint contribution</td>
</tr>
</tbody>
</table>

Algorithm:
1. enter shortbread sold, pecan sandies sold, and chocolate mint sold
2. calculate the total sold by adding together shortbread sold, pecan sandies sold, and chocolate mint sold
3. calculate shortbread contribution by dividing shortbread sold by total sold, and then multiplying the result by 100
4. calculate pecan sandies contribution by dividing pecan sandies sold by total sold, and then multiplying the result by 100
5. calculate chocolate mint contribution by dividing chocolate mint sold by total sold, and then multiplying the result by 100
6. display shortbread contribution, pecan sandies contribution, and chocolate mint contribution

### LAB 2-5  Desk-Check

<table>
<thead>
<tr>
<th>restaurant bill</th>
<th>sales tax</th>
<th>bill before sales tax</th>
<th>10% tip</th>
<th>15% tip</th>
<th>20% tip</th>
</tr>
</thead>
<tbody>
<tr>
<td>102.50</td>
<td>5.80</td>
<td>96.70</td>
<td>9.67</td>
<td>14.51</td>
<td>19.34</td>
</tr>
<tr>
<td>56.78</td>
<td>2.18</td>
<td>54.60</td>
<td>5.46</td>
<td>8.19</td>
<td>10.92</td>
</tr>
</tbody>
</table>

### LAB 2-6  Debug

<table>
<thead>
<tr>
<th>first number</th>
<th>second number</th>
<th>third number</th>
<th>sum</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>56</td>
<td>70</td>
<td>159</td>
<td>53</td>
</tr>
</tbody>
</table>
Answers to Chapter 3 Mini-Quizzes

Mini-Quiz 3-1
1. one
2. a. salesTax
3. d. TAX_RATE
4. variables and named constants

Mini-Quiz 3-2
1. a. True
2. b. 11011
3. 72, 01001000
4. d. both a and c

Mini-Quiz 3-3
1. a. ‘$’
2. d. all of the above
3. 3
4. int population = 0;
5. const double INTEREST_RATE = 0.05;
6. $5

Answers to Chapter 3 Labs

LAB 3-1 Stop and Analyze
1. The problem requires five memory locations.
2. The problem requires one named constant for the item cost. A named constant is appropriate because the item cost will always be $5.45. The remaining four input, processing, and output items require variables. Variables were chosen for these items so that their values can vary during runtime.
3. int quantitySold = 0;
   const double ITEM_COST = 5.45;
   double sellingPrice = 0.0;
   double difference = 0.0;
   double profit = 0.0;

LAB 3-2 Plan and Create

No answer required.

LAB 3-3 Modify

The modifications made to Figures 3-18 and 3-21 are shaded.

<table>
<thead>
<tr>
<th>Input</th>
<th>Processing</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>commission rate</td>
<td>Processing items: none</td>
<td>commission</td>
</tr>
<tr>
<td>sales amount</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Algorithm:
1. enter the commission rate and sales amount
2. calculate the commission by multiplying the sales amount by the commission rate
3. display the commission

IPO chart information

Input

commission rate
sales amount

Processing
none

Output
commission

C++ instructions

double CommRate = 0.0;
double sales = 0.0;

double commission = 0.0;

LAB 3-4 What’s Missing?

The missing statement is shaded.

<table>
<thead>
<tr>
<th>IPO chart information</th>
<th>C++ instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>height</td>
<td>double height = 0.0;</td>
</tr>
<tr>
<td>radius</td>
<td>double radius = 0.0;</td>
</tr>
<tr>
<td>pi (3.14)</td>
<td>const double PI = 3.14;</td>
</tr>
</tbody>
</table>
**Answers to Mini-Quizzes and Labs**

**Processing**
none

**Output**
volume double volume = 0.0;

---

**LAB 3-5  Desk-Check**

Algorithm:
1. enter the height and radius
2. calculate the volume by multiplying the radius by itself, and then
   multiplying the result by pi, and then multiplying that result by the height
3. display the volume

<table>
<thead>
<tr>
<th>height</th>
<th>radius</th>
<th>pi</th>
<th>volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>15</td>
<td>3.14</td>
<td>12010.50</td>
</tr>
</tbody>
</table>

---

**LAB 3-6  Debug**

The modifications made to Figure 3-23 are shaded in the C++ instructions column.

**IPO chart information**

**Input**
- first number
- second number
- third number

**Processing**
- sum double sum = 0.0;

**Output**
- average double average = 0.0;

**C++ instructions**
- double first = 0.0;
- double second = 0.0;
- double third = 0.0;

---

**Answers to Chapter 4 Mini-Quizzes**

**Mini-Quiz 4-1**

1. a. cin >> population;
2. d. cout << quantity;
3. d. all of the above
4. <<
Mini-Quiz 4-2

1. `totalDue = 2.5 * quantity;`
2. `totalDue = 2.5 * static_cast<double>(quantity);`
3. The expression will evaluate to 5.5. It should evaluate to 6. The expression evaluates incorrectly because dividing the integer 9 by the integer 2 results in the integer 4 rather than in the `double` number 4.5. Adding 1.5 to 4 results in the incorrect answer of 5.5.
4. You can use any of the following expressions. You can also use the `static_cast` operator to type cast at least one of the integers in the expression.
   - `9.0 / 2.0 + 1.5`
   - `9.0 / 2 + 1.5`
   - `9 / 2.0 + 1.5`
5. `ordered -= 7;`

Mini-Quiz 4-3

1. syntax
2. source
3. braces

Answers to Chapter 4 Labs

**LAB 4-1 Stop and Analyze**

1. 1. The `quantity` integer (10) is *implicitly* promoted to the `double` number 10.0.
   2. The result of Step 1 (10.0) is multiplied by the `double` number stored in the `itemCost` variable (5.35), giving 53.5.
   3. The integer 5 is *implicitly* promoted to the `double` number 5.0
   4. The result of Step 2 (53.5) is added to the result of Step 3 (5.0), giving 58.5.
   5. The result of Step 4 (58.5) is assigned to the `amountDue` variable. The value is correct.

2. 1. The `total` integer (30) is divided by the integer 3, giving 10.
   2. The result of Step 1 (10) is assigned to the `total` variable. The value is correct.
3. 1. The integer 2 is implicitly promoted to the double number 2.0.
2. The result of Step 1 (2.0) is divided into the double number stored in the store2Sales variable (8325.72), giving 4162.86.
3. The result of Step 2 (4162.86) is added to the double number stored in the store1Sales variable (5678.43), giving 9841.29.
4. The result of Step 3 is assigned to the avgSales variable. The value is not correct. To fix the assignment statement, include parentheses around the addition operation, like this: avgSales = (store1Sales + store2Sales) / 2;

4. 1. The midterm integer (74) is added to the final integer (93), giving 167.
2. The result of Step 1 (167) is divided by 2, giving 83.
3. The result of Step 2 is assigned to the average variable. The value is not correct. Two ways you can fix the statement are shown here:

```cpp
average = (midterm + final) / 2.0;
average = static_cast<double>(midterm + final) / 2;
```

---

**LAB 4-2 Plan and Create**
No answer required.

**LAB 4-3 Modify**
The modifications made to the Lab4-2.cpp file are shaded.

```cpp
1 //Lab4-3.cpp - displays a salesperson's commission
2 //Created/revised by <your name> on <current date>
3 #include <iostream>
4 using namespace std;
5
6 int main()
7 {
8     //declare variables
9     double commRate = 0.0;
10    double sales = 0.0;
11    double commission = 0.0;
12    double commRate = 0.0;
13
14    //enter input items
15    cout << "Sales amount: ";
16
17    //calculate commission
18    commission = sales * commRate;
19
20    //display output
21    cout << "Commission: 
22```
```
**Answers to Chapter 4 Labs**

```cpp
16    cin >> sales;
17    cout << "Commission rate (in decimal form): ";
18    cin >> commRate;
19
20    //calculate and display the commission
21    commission = sales * commRate;
22    cout << "Commission: 
23         \$" << commission << endl;
24
25    return 0;
26 } //end of main function
```

---

**LAB 4-4  What's Missing?**

//Lab4-4.cpp - displays the volume of a cylinder
//Created/revised by <your name> on <current date>

```cpp
#include <iostream>
using namespace std;

int main()
{
    const double PI = 3.14;
    double height   = 0.0;
    double radius   = 0.0;
    double volume   = 0.0;

    cout << "Height: ";
    cin >> height;
    cout << "Radius: ";
    cin >> radius;

    volume = PI * radius * radius * height;
    cout << "Volume: 
         \$" << volume << endl;

    return 0;
} //end of main function
```

---

**LAB 4-5  Desk-Check**

<table>
<thead>
<tr>
<th>num</th>
<th>result1</th>
<th>result2</th>
<th>result3</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>37</td>
<td>37.5</td>
<td></td>
</tr>
</tbody>
</table>
LAB 4-6  Debug

To debug the program, you need to convert at least one of the items on the right side of the assignment operator to the double data type. For example, you can use any of the following:

```cpp
area = base * height / 2.0;
area = static_cast<double>(base) * height / 2;
area = base * static_cast<double>(height) / 2;
area = static_cast<double>(base) * static_cast<double>(height) / 2.0;
```

Answers to Chapter 5 Mini-Quizzes

Mini-Quiz 5-1

1. end if
2. b. False
3. a. diamond
4. b. single-alternative
5. a. dual-alternative

Mini-Quiz 5-2

1. a. braces
2. d. if (age == 21)
3. b. if (price >= 12.75)
4. c. !=
5. b. >=

Mini-Quiz 5-3

1. false
2. true
3. false
4. b. if (age >= 30 && age <= 40)
5. a. if (code == 'R' || code == 'r')
Mini-Quiz 5-4

1. d. `cout << fixed << setprecision(2);`
2. c. `letter = tolower(letter);`
3. d. 34.650000

Answers to Chapter 5 Labs

** LAB 5-1  Stop and Analyze **

1. The 0.2 rate will be assigned to the `rate` variable when the user enters either code 1 or code 3. The 0.15 rate will be assigned to the `rate` variable when the user enters one of the following codes: 2, 4, and 5.

2. The directive on Line 5 is necessary because the program uses the `setprecision` stream manipulator.

3. The literal constants on Line 19 are enclosed in single quotation marks because the `code` variable's data type is `char`.

4. ```
   if (code != '1' && code != '3')
       rate = 0.15;
   else
       rate = 0.2;
   //end if
   ```

5. You can also write the statement on Line 26 as follows:
   ```
   sales = sales * (1 + rate);
   sales *= (1 + rate);
   sales += (sales * rate);
   ```

6. To omit the `rate` variable from the program, you would need to remove the `double rate = 0.0;` and `sales = sales + sales * rate;` statements. You would also need to change the `rate = 0.2;` statement to `sales = sales + sales * 0.2;`, and change the `rate = 0.15;` statement to `sales = sales + sales * 0.15;`.

** LAB 5-2  Plan and Create **

No answer required.
LAB 5-3  Modify

The changes made to the original program are shaded in the following partial program.

```c
int main()
{
    const double MEMBER_DISCOUNT_RATE = 0.1;
    const double NONMEMBER_DISCOUNT_RATE = 0.05;
    const double SHIP_CHG1 = 0.99;
    const double SHIP_CHG2 = 4.99;
    double amtOwed = 0.0;
    char member = ' ';
    double discount = 0.0;

    // enter input items
    cout << "Amount owed before any discount and shipping: ";
    cin >> amtOwed;
    cout << "Premier Club member (Y/N)? ";
    cin >> member;

    // subtract discount
    if (toupper(member) == 'Y')
        discount = amtOwed * MEMBER_DISCOUNT_RATE;
    else
        discount = amtOwed * NONMEMBER_DISCOUNT_RATE;
    // end if
    amtOwed -= discount;
}
```

LAB 5-4  What's Missing?

//Lab5-4.cpp - displays the total due for tickets
//Created/revised by <your name> on <current date>

```c
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    const int TICKET_PRICE = 75;
    int tickets = 0;
    int total = 0;
    cout << "Number of tickets you want to purchase (the maximum is 10): ";
    cin >> tickets;
    if (tickets > 0 && tickets < 11)
    {
        total = tickets * TICKET_PRICE;
        cout << fixed << setprecision(0);
        cout << "Price: $" << total << endl;
    }
    else
    {
        cout << "Invalid number of tickets." << endl;
        // end if
    }
    return 0;
    // end of main function
```
When the user enters the number 0, the condition in the first if statement evaluates to true and the statement’s true path displays the message “The quantity must be greater than 0.” Although the correct message already appears on the screen, the computer still evaluates the second if statement’s condition, which determines whether the quantity is greater than 0. The second evaluation is unnecessary and makes the code inefficient. You can fix the code by replacing the first //end if comment and the second if clause with an else clause.

To debug the program, change if (code = '2') to if (code == '2').

Answers to Chapter 6 Mini-Quizzes

Mini-Quiz 6-1

1. c. in either an outer selection structure’s false path or its true path

2. 1. shoot the basketball
   2. if (the basketball did not go through the hoop)
      say “Missed it!”
      if (the basketball hit the rim)
      say “So close”
      end if
   else
   say “I did it!”
   if (Maleek was behind the 3-point line)
   say “3 points for me”
   else
   say “2 points for me”
   end if
   end if

3. ask the store clerk whether the store accepts your credit card
   if (the store accepts your credit card)
   pay for your items using your credit card
   else
   ask the store clerk whether the store accepts your debit card
if (the store accepts your debit card)
    pay for your items using your debit card
else
    pay for your items using cash
end if

end if

Mini-Quiz 6-2

1. if (the test score is at least 90)
    display "Great score!"
else
    if (the test score is at least 70)
        display "Good score"
    else
        display "Retake the test"
    end if
end if

2.

3. if (score >= 90)
    cout << "Great score!" << endl;
else
    if (score >= 70)
        cout << "Good score" << endl;
    else
        cout << "Retake the test" << endl;
        //end if
    //end if

4. a. membership status, day of the week
Mini-Quiz 6-3

1. Using a compound condition rather than a nested selection structure
   Reversing the decisions in the outer and nested selection structures
   Using an unnecessary nested selection structure
   Including an unnecessary comparison in a condition

2. Algorithm A in Figure 6-19 is inefficient because the nested selection structure is
   unnecessary. If the first selection structure's condition evaluates to false, it means
   that the basketball did not go through the hoop. Therefore, there is no need for a
   nested selection structure that determines whether the basketball did not go through
   the hoop.

3. Algorithm B in Figure 6-19 will not give you the same results as the algorithm shown
   in Figure 6-2. Algorithm B has Maleek saying "I did it!" and "3 points for me" even
   when the basketball does not go through the hoop.

4. Algorithm C in Figure 6-19 will give you the same results as the algorithm shown in
   Figure 6-2.

Mini-Quiz 6-4

1. if (score >= 90)
   
   cout << "Great score!" << endl;
   
   else if (score >= 70)
     
     cout << "Good score" << endl;
   
   else if (score >= 0)
     
     cout << "Retake the test" << endl;
   
   else
     
     cout << "Invalid test score" << endl;
     //end if

2. b. case 'B':

3. break

Answers to Chapter 6 Labs

LAB 6-1 Stop and Analyze

1. The program will display the number 11.46 when the ID is the number 11.

2. if (id == 1)
   
   price = 50.55;
   
   else
     
     if (id == 2 || id == 9)
       
       price = 12.35;
     
     else
       
       if (id == 5 || id == 7 || id ==11)
         
         price = 11.46;
       
     else
       
       // end if
```cpp
int price = -1

3. if (id == 1)
    price = 50.55;
else if (id == 2 || id == 9)
    price = 12.35;
else if (id == 5 || id == 7 || id == 11)
    price = 11.46;
else
    price = -1;

4. switch (id)
{
    case 1:
        price = 50.55;
        break;
    case 2:
    case 9:
        price = 12.35;
        break;
    case 5:
    case 7:
    case 11:
        price = 11.46;
        break;
    default:
        price = -1;
}  //end switch

5. switch (id)
{
    case 1:
        price = 50.55;
        cout << price << endl;
        break;
    case 2:
    case 9:
        price = 12.35;
        cout << price << endl;
        break;
    case 5:
    case 7:
    case 11:
        price = 11.46;
        cout << price << endl;
        break;
    default:
        cout << "Invalid ID" << endl;
}  //end switch
```
LAB 6-2  Plan and Create
No answer required.

LAB 6-3  Modify
//Lab6-3.cpp - displays the price of a pizza
//Created/revised by <your name> on <current date>

#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    char size = ' ';
    char coupon = ' ';
    double price = 0.0;

    cout << "M(edium) or L(arge) pizza? ";
    cin >> size;
    size = toupper(size);
    if (size != 'M' && size != 'L')
        cout << "Please enter either M or L." << endl;
    else
    {
        if (size == 'M')
            price = 9.99;
        else
            //large pizza
            price = 12.99;
        //end if
        cout << "$2 coupon (Y/N)? ";
        cin >> coupon;
        if (toupper(coupon) == 'Y')
            price -= 2;
        //end if
        cout << fixed << setprecision(2);
        cout << "Price: $" << price << endl;
    } //end if

    return 0;
} //end of main function
LAB 6-4  What's Missing?

//Lab6-4.cpp - displays the price of a ticket
//Created/revised by <your name> on <current date>

#include <iostream>
using namespace std;

int main()
{
    int age = 0;
    int price = 0;

    cout << "Age (years): " << age;
    cin >> age;

    if (age < 0)
        cout << "Invalid age" << endl;
    else
    {
        if (age > 64)
            price = 6;
        else if (age > 3)
            price = 9;
        else
            price = 0;
        //end if
        cout << "Price: $" << price << endl;
    }  //end if
    return 0;
}   //end of main function

LAB 6-5  Desk-Check

number

0
2
4
0
5
10
0
100
50
// Lab 6-6 Debug
The modifications are shaded in the code.

#include <iostream>
using namespace std;

int main()
{
    int code = 0;
    cout << "Enter the code (1 through 5): ";
    cin >> code;

    // Display salary
    if (code == 1)
        cout << "$45,000" << endl;
    else if (code == 2 || code == 5)
        cout << "$33,000" << endl;
    else if (code == 3 || code == 4)
        cout << "$25,000" << endl;
    else
        cout << "Entry error" << endl;
    // end if
    return 0;
} // end of main function

---

Answers to Chapter 7 Mini-Quizzes

Mini-Quiz 7-1

1. 1. print the first page
    2. repeat while (there is another page to print)
       print the next page
       end repeat
    say "Done printing"
2. 1. print the first page  
   2. repeat while (there is another page to print)  
      if (the next page is not page 3)  
         print the next page  
      end if  
   end repeat  
   say “Done printing”  

3. 1. shoot the basketball  
   2. repeat while (the basketball did not go through the hoop)  
      say “Missed it!”  
      shoot the basketball  
   end repeat  
   3. say “I did it!”

Mini-Quiz 7-2

1. while (ordered > 100)  
2. while (quantity >= 0)  
3. while (inStock > reorder)  
4. while (toupper(letter) == 'Y') [You can also use while (tolower(letter) == 'y') or while (letter == 'Y' || letter == 'y').]  
5. a. -9

Mini-Quiz 7-3

1. b. counter  
2. quantity += 10; (or quantity = quantity + 10;)
3. total -= 5; (or total = total - 5; or total += -5; or total = total + -5;)
4. totalPurchases += purchases; (or totalPurchases = totalPurchases + purchases;)
Mini-Quiz 7-4

1. while (evenNum < 9)
   {
      cout << evenNum << endl;
      evenNum += 2;  // (or evenNum = evenNum + 2;)
   } // end while

2. a. for (int x = 10; x <= 100; x = x + 10)

3. d. 110

4. for (int x = 25; x > 0; x = x – 5) (You can also use x -= 5, x += -5, or
   x = x + -5 as the update argument.)

5. 0

6. for (int num = 2; num < 9; num += 2) (You can also use num = num + 2 as the
   update argument.)

Answers to Chapter 7 Labs

LAB 7-1  Stop and Analyze

1. The program’s counter and accumulator variables are named day and
   totalTexts, respectively.

2. The for clause’s condition argument can also be phrased as day <= 7.

3. The for statement’s counter variable is declared at the beginning of the program
   because it needs to be used after the for loop ends. More specifically, it needs to
   be used by the statement on Line 24, which calculates the average number of text
   messages.

4. The average number of text messages is 74.

<table>
<thead>
<tr>
<th>day</th>
<th>dailyTexts</th>
<th>totalTexts</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>76</td>
<td>76</td>
<td>74.0</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>256</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>43</td>
<td>299</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>68</td>
<td>367</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>70</td>
<td>437</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>79</td>
<td>516</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. The program displays 74.
6. The build or compile operation was not successful because the compiler does not recognize the day variable used in Line 24. When the for loop ended, that variable was removed from the computer’s internal memory.

8. The program does not work correctly because it prompts the user to enter the number of text messages for days from 0 through 7, rather than from 1 through 7. To fix the program change the day variable’s declaration statement to `int day = 1;`.

LAB 7-2 Plan and Create

No answer required.

LAB 7-3 Modify

//Lab7-3.cpp - calculates the average number of text messages sent each day for 7 days
//Created/revised by <your name> on <current date>

#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int day = 1;
    int totalTexts = 0;
    int dailyTexts = 0;
    double average = 0.0;

    while (day < 8)
    {
        cout << "How many text messages did you send on day "
             << day << "? ";
        cin >> dailyTexts;
        totalTexts += dailyTexts;
        day += 1;
    } //end while

    average = static_cast<double>(totalTexts) / (day - 1);
    cout << fixed << setprecision(0);
    cout << endl << "You sent approximately "
         << average << " text messages per day."
         << endl;
    return 0;
} //end of main function
LAB 7-4  What's Missing?
//Lab7-4.cpp - displays the average electric bill
//Created/revised by <your name> on <current date>

#include <iostream>
#include <iomanip>

using namespace std;

int main()
{
    double bill = 0.0;
    double totalBills = 0.0;
    double avgBill = 0.0;
    int months = 0;

    cout << "Bill for month 1: ";
    cin >> bill;
    while (bill >= 0.0)
    {
        totalBills += bill;
        months += 1;
        cout << "Bill for month " << months + 1 << ": ";
        cin >> bill;
    }  //end while

    if (months > 0)
    {
        avgBill = totalBills / months;
        cout << fixed << setprecision(2);
        cout << "Average electric bill for " << months << " months: $" << avgBill << endl;
    }
    else
        cout << "No bill amount entered." << endl;
    //end if
    return 0;
}  //end of main function

LAB 7-5  Desk-Check

The original code contains an error because it will not display the number 12. To correct the code, change the for statement's condition to number <= 12.

First desk-check:   Corrected desk-check:
number  number
1       1
2       2
4       4
8       8
16      16
12      12
14      14
To debug the program, enter a `cin >> price;` statement below the `cout << "Next price: ";` statement.

Answers to Chapter 8 Mini-Quizzes

**Mini-Quiz 8-1**

1. The condition will be evaluated four times. The algorithm will display the numbers 5, 8, and 9, followed by the word Done.
2. The condition will be evaluated once. The algorithm will display the word Done.
3. The condition will be evaluated three times. The algorithm will display the numbers 5, 8, and 9, followed by the word Done.
4. The condition will be evaluated once. The algorithm will display the number 0 followed by the word Done.

**Mini-Quiz 8-2**

1. a. `do`
2. d. semicolon
3. `} while (ordered <= inventory);`
4. `} while (toupper(letter) == 'Y');`

**Mini-Quiz 8-3**

1. b. False
2. b. nested, outer
3. a. outer, nested
**Answers to Chapter 8 Labs**

**LAB 8-1  Stop and Analyze**

1. The program contains three loops. Two of the loops are nested.

2. Desk-check:

<table>
<thead>
<tr>
<th>maxRows</th>
<th>row</th>
<th>space</th>
<th>asterisk</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

   The program will display the following pattern of asterisks:

   * 
   * * 
   * * * 

3. The program displays the pattern of asterisks shown above.

4. The program displays the following pattern of asterisks:

   * 
   * * 
   * * * 
   * * * * 
   * * * * * 
   * * * * * * 
   * * * * * * * 
   * * * * * * * * 
   * * * * * * * * * 
   * * * * * * * * * * 

5. //Lab8-1.cpp
   //Created/revised by <your name> on <current date>

   ```cpp
   #include <iostream>
   using namespace std;

   int main()
   {
      int maxRows = 0;
      int row = 0;

      cout << "Maximum number of rows: ";
      cin >> maxRows;

      while (row < maxRows)
      {
         for (int space = 0; space < maxRows - row; space += 1)
            cout << " ";

         //end for
         cout << endl;
      }
   }
   ```
for (int asterisk = 0; asterisk <= row; asterisk += 1)
    cout << "* ";
//end for
    cout << endl;
row += 1;
} //end while
return 0;
} //end of main function

LAB 8-2  Plan and Create
No answer required.

LAB 8-3  Modify
//Lab8-3.cpp - displays a person's total earnings before retirement at age 65,
//using annual raise rates of 3%, 4%, and 5%.
//Created/revised by <your name> on <current date>
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int age = 0;
    int currentSalary = 0;
    int yearsToRetire = 0;
    double newSalary = 0.0;
    double total = 0.0;
    double rate = 0.0;

    cout << fixed << setprecision(0);
    cout << "Current age in years (1 to 64): ";
    cin >> age;

    if (age < 1 || age > 64)
        cout << "Please enter an age from 1 to 64."

    else
    {
        cout << "Current salary as a whole number: ";
        cin >> currentSalary;
        cout << endl;

        yearsToRetire = 65 - age;
        rate = 0.03;
        do
        {
            newSalary = currentSalary; //year 1 salary
            total = currentSalary;     //year 1 salary
            currentSalary = newSalary * (1 + rate);
            total += newSalary;
            cout << "Year ";
            cout << num << " current salary: ";
            cout << currentSalary;
            cout << " total earned: ";
            cout << total;
            cout << endl;
        }
        while (yearsToRetire > 0);
        yearsToRetire--;
    }
    return 0;
} //end of main function
for (int year = 2; year <= yearsToRetire; year += 1)
{
    newSalary *= (1 + rate);
    total += newSalary;
} // end for
cout << "Total with a " << rate * 100
<< " % annual raise: $" << total << endl;
rate += 0.01;
} while (rate < 0.06);
} // end if
return 0;
} // end of main function

**LAB 8-4  What's Missing?**

//Lab8-4.cpp - displays a pattern of numbers
//Created/revised by <your name> on <current date>

#include <iostream>
using namespace std;

int main()
{
    int maxRows = 0;
cout << "How many rows? ";
cin >> maxRows;

for (int row = 1; row <= maxRows; row += 1)
{
    for (int col = 1; col <= row; col += 1)
    {
        cout << col;
    } // end for
    cout << endl;
} // end for
return 0;
} // end of main function

**LAB 8-5  Desk-Check**

Desk-check:

<table>
<thead>
<tr>
<th>sumX</th>
<th>sumY</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>16</td>
<td>12</td>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td>18</td>
<td>14</td>
<td>18</td>
<td>14</td>
</tr>
</tbody>
</table>

The program will display the following:
sumX value: 12
sumY value: 12

When the outer loop ends, the x variable contains the number 8. When the nested loop ends, the y variable contains the number 5.

**LAB 8-6 Debug**

To debug the program, cut the `month += 1;` statement from the nested loop and paste it below the `cout << endl;` statement in the outer loop. Also, change the `totalSales += totalSales + sales;` statement to either `totalSales = totalSales + sales;` or `totalSales += sales;`.

**Answers to Chapter 9 Mini-Quizzes**

**Mini-Quiz 9-1**

1. b. `sqrt(16.0)`  
2. c. `25 + rand() % (50 - 25 + 1)`  
3. d. none of the above  
4. a. `#include <ctime>`

**Mini-Quiz 9-2**

1. d. all of the above  
2. a. `double getArea()`  
3. `double getGrossPay(int hours, double rate)`  
4. `return gross;`
Mini-Quiz 9-3

1. b. area = getArea();
2. c. double getArea();
3. cout << getArea();
4. double getGrossPay(int hours, double rate); or double getGrossPay(int, double);
5. weekGross = getGrossPay(40, payRate);

Answers to Chapter 9 Labs

LAB 9-1 Stop and Analyze

1. The statements on Lines 9 and 10 are function prototypes, and they are necessary because the getArea and getDiameter functions are defined below the main function.
2. The statement on Line 9 could also be written as follows: double getArea(double);
3. Without the outer selection structure, the program will display the following: Diameter: 20.
4. The scope of the choice and radius variables is the main function (Lines 17 through 36). The lifetime is the same as the main function's lifetime, which means that the variables will be removed from the computer's memory when the main function ends.
5. The scope of the rad variable used in the getArea function is the function itself (Lines 40 through 45). Its lifetime is the same as the getArea function's lifetime, which means it will be removed from the computer's memory when the function ends.
6. The scope of the PI constant and the area variable is the getArea function (Lines 43 through 45). Their lifetime is the same as the getArea function's lifetime, which means they will be removed from the computer's memory when the function ends.
7. The scope of the rad variable used in the getDiameter function is the function itself (Lines 48 through 50). Its lifetime is the same as the getDiameter function's lifetime, which means it will be removed from the computer's memory when the function ends.
9. //Lab9-1.cpp – circle calculations
   //Created/revised by <your name> on <current date>

   #include <iostream>
   #include <cmath>
   using namespace std;

   //function prototypes
   double getArea(double rad);
   double getDiameter(double rad);
   double getCircumference(double rad);
int main()
{
    int choice = 0;
    double radius = 0.0;

    cout << "1   Circle area" << endl;
    cout << "2   Circle diameter" << endl;
    cout << "3   Circle circumference" << endl;
    cout << "Enter your choice (1, 2, or 3): ";
    cin >> choice;

    if (choice < 1 || choice > 3)
        cout << "Invalid choice" << endl;
    else
    {
        cout << "Radius: ";
        cin >> radius;
        if (choice == 1)
            cout << "Area: " << getArea(radius);
        else if (choice == 2)
            cout << "Diameter: " << getDiameter(radius);
        else
            cout << "Circumference: " << getCircumference(radius);
        //end if
        cout << endl;
    }  //end if
    return 0;
}   //end of main function

//*****function definitions*****
double getArea(double rad)
{
    const double PI = 3.141593;
    double area = 0.0;
    area = PI * pow(rad, 2);
    return area;
}  //end getArea function

double getDiameter(double rad)
{
    return 2 * rad;
}  //end getDiameter function

double getCircumference(double rad)
{
    const double PI = 3.141593;
    double circumference = 0.0;
    circumference = 2 * PI * rad;
    return circumference;
}  //end getCircumference function
LAB 9-2 Plan and Create

No answer required.

LAB 9-3 Modify

//Lab9-3.cpp - displays two monthly car payments
//Created/revised by <your name> on <current date>

#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

//function prototype
double getPayment(int, double, int);

int main()
{
    int carPrice = 0;
    int rebate = 0;
    double creditRate = 0.0;
    double dealerRate = 0.0;
    int term = 0;
    double creditPayment = 0.0;
    double dealerPayment = 0.0;
    char another = 'Y';

    while (another == 'Y')
    {
        cout << "Car price (after any trade-in): ";
        cin >> carPrice;
        cout << "Rebate: ";
        cin >> rebate;
        cout << "Credit union rate: ";
        cin >> creditRate;
        cout << "Dealer rate: ";
        cin >> dealerRate;
        cout << "Term in years: ";
        cin >> term;

        if (creditRate >= 1)
            creditRate /= 100;
        //end if
        if (dealerRate >= 1)
            dealerRate /= 100;
        //end if
    }
}

// function getPayment

double getPayment(int carPrice, double creditRate, int term)
{
    // calculation logic
    return creditPayment;
}

// function getPayment

double getPayment(int carPrice, double dealerRate, int term)
{
    // calculation logic
    return dealerPayment;
}
//call function to calculate payments
creditPayment = getPayment(carPrice - rebate,
    creditRate / 12, term * 12);
dealerPayment = getPayment(carPrice,
    dealerRate / 12, term * 12);

//display payments
cout << fixed << setprecision(2) << endl;
cout << "Credit union payment: $" << creditPayment << endl;
cout << "Dealer payment: $" << dealerPayment << endl;

if (creditPayment < dealerPayment)
c    cout << "Take the rebate and finance through the credit union.";
else if (creditPayment > dealerPayment)
c    cout << "Don't take the rebate. Finance through the dealer.";
else
c    cout << "You can finance through the dealer or the credit union.";
//end if

cout << endl;
cout << "Calculate another set of payments (Y/N)?";
cin >> another;
another = toupper(another);
} //end while

return 0;
} //end of main function

//*****function definitions*****
double getPayment(int prin,
    double monthRate,
    int months)
{
    //calculates and returns a monthly payment
    double monthPay = 0.0;
    monthPay = prin * monthRate /
               (1 - pow(monthRate + 1, -months));
    return monthPay;
} //end of getPayment function
LAB 9-4 What's Missing?

//Lab9-4.cpp - displays total due
//Created/revised by <your name> on <current date>

#include <iostream>
#include <iomanip>
using namespace std;

double getSalesTax(double subtotal);

int main()
{
    int quantity = 0;
    double itemPrice = 0.0;
    double totalBeforeTax = 0.0;
    double totalDue = 0.0;

    cout << "Quantity: ";
    cin >> quantity;
    cout << "Item price: ";
    cin >> itemPrice;
    totalBeforeTax = quantity * itemPrice;
    totalDue = totalBeforeTax + getSalesTax(totalBeforeTax);

    cout << fixed << setprecision(2) << endl;
    cout << "Total due: $" << totalDue << endl;
    return 0;
} //end of main function

//*****function definitions*****
double getSalesTax(double subtotal)
{
    const double TAX_RATE = 0.1;
    double tax = 0.0;
    tax = subtotal * TAX_RATE;
    return tax;
} //end of getSalesTax function
**LAB 9-5  Desk-Check**

Desk-check:

<table>
<thead>
<tr>
<th>currentTotal</th>
<th>number</th>
<th>x</th>
<th>current</th>
<th>num</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>+</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>100</td>
<td>-1</td>
<td>+</td>
<td>100</td>
<td>+</td>
</tr>
<tr>
<td>-3</td>
<td>-1</td>
<td>-</td>
<td>-3</td>
<td>-1</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>+</td>
<td>100</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>-3</td>
<td>-</td>
<td>3</td>
<td>-3</td>
</tr>
<tr>
<td>110</td>
<td>2</td>
<td>+</td>
<td>110</td>
<td>+</td>
</tr>
<tr>
<td>-5</td>
<td>-5</td>
<td>-</td>
<td>-5</td>
<td>-5</td>
</tr>
<tr>
<td>210</td>
<td>1</td>
<td>+</td>
<td>210</td>
<td>+</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>-</td>
<td>10</td>
<td>-1</td>
</tr>
<tr>
<td>281</td>
<td>-5</td>
<td>+</td>
<td>281</td>
<td>+</td>
</tr>
</tbody>
</table>

The code will display 281 as the current total.

**LAB 9-6  Debug**

To debug the program, change the statement that calls the `getDepreciation` function to `depreciation = getDepreciation(cost, salvage, lifeYears);`.

---

**Answers to Chapter 10 Mini-Quizzes**

**Mini-Quiz 10-1**

1. `void`
2. `displayTaxes(federalTax, localTax);`
3. `void displayTaxes(double fedTax, double stateTax)`
4. b. False

**Mini-Quiz 10-2**

1. `void calcTaxes(double pay, double &fedTax, double &stateTax)`
2. `calcTaxes(gross, federal, state);`
3. `void calcTaxes(double pay, double &fedTax, double &stateTax); or void calcTaxes(double, double &, double &)`
4. b. False
Answers to Chapter 10 Labs

LAB 10-1  Stop and Analyze

1. Lines 10 and 52 indicate whether the variables are passed by value or by reference to the getArea function.

2. The radius variable is passed by value because the receiving functions need to know its value but do not need to change its contents. The circleArea and circleDiameter variables are passed by reference because the receiving functions need to store values (either the area or the diameter) inside the variables.

3. The displayChoices function is a void function because it does not need to return a value after completing its task.

4. The getArea and getDiameter functions use the variables that are passed by reference to send information back to the main function.

5. //Lab10-1.cpp - circle calculations
   //Created/revised by <your name> on <current date>

   #include <iostream>
   #include <cmath>
   using namespace std;

   //function prototypes
   void displayChoices();
   double getArea(double rad);
   void getDiameter(double rad, double &diameter);

   int main()
   {
   int choice = 0;
   double radius = 0.0;
   double circleArea = 0.0;
   double circleDiameter = 0.0;

   displayChoices();
   cout << "Enter your choice (1 or 2): ";
   cin >> choice;

   if (choice < 1 || choice > 2)
      cout << "Invalid choice" << endl;
   else
   {
   cout << "Radius: ";
   cin >> radius;
   if (choice == 1)
   {
   circleArea = getArea(radius);
   cout << "Area: " << circleArea;
   }
else
{
    getDiameter(radius, circleDiameter);
    cout << "Diameter: " << circleDiameter;
} //end if
    cout << endl;
} //end if
return 0;
} //end of main function

****function definitions****
void displayChoices()
{
    cout << "1 Circle area" << endl;
    cout << "2 Circle diameter" << endl;
} //end displayChoices

double getArea(double rad)
{
    const double PI = 3.141593;
    double area = 0.0;

    area = PI * pow(rad, 2);
    return area;
} //end getArea function

void getDiameter(double rad, double &diameter)
{
    diameter = 2 * rad;
} //end getDiameter function

---

**LAB 10-2** **Plan and Create**

No answer required.

---

**LAB 10-3** **Modify**

//Lab10-3.cpp - displays total owed
//Created/revised by <your name> on <current date>

#include <iostream>
#include <iomanip>
using namespace std;
/function prototypes
void displayOptions();
double getRegular(int windows, double price);
double getBoGo(int windows, double price);

int main()
{
    int option = 0;
    int numOrdered = 0;
    double winPrice = 0.0;
    double totalOwed = 0.0;

    cout << fixed << setprecision(2);

displayOptions();
cout << "Pricing option? ";
cin >> option;

if (option == 1 || option == 2)
{
    cout << "Number of windows: ";
cin >> numOrdered;
    cout << "Price per window: " ;
cin >> winPrice;

    if (option == 1)
    {
        totalOwed = getRegular(numOrdered, winPrice);
    }
    else
    {
        totalOwed = getBoGo(numOrdered, winPrice);
    }

    cout << "Total owed----- $" << totalOwed << endl << endl;
}
else
    cout << "Invalid option" << endl;

return 0;
} //end of main function

//*****function definitions*****
void displayOptions()
{
    cout << "Pricing options:" << endl;
    cout << "1 Regular pricing" << endl;
    cout << "2 BOGO pricing" << endl;
} //end displayOptions
double getRegular(int windows, double price) {
    double total = 0.0;
    total = windows * price;
    return total;
}  //end getRegular function

double getBoGo(int windows, double price) {
    double total = 0.0;
    total = (windows / 2 + windows % 2) * price;
    return total;
}  //end getBoGo function

LAB 10-4 What's Missing?

//Lab10-4.cpp - displays the raise and new salary
//Created/revised by <your name> on <current date>

#include <iostream>
#include <iomanip>
using namespace std;

//function prototypes
void getRaise(double current, double rate, double &increase);
void getNewPay(double current, double increase, double &newPay);

int main() {
    double currentSalary = 0.0;
    double raiseRate = 0.0;
    double raise = 0.0;
    double newSalary = 0.0;

    cout << "Current salary: ";
    cin >> currentSalary;
    cout << "Raise rate (in decimal form): ";
    cin >> raiseRate;

    //get the raise and new salary
    getRaise(currentSalary, raiseRate, raise);
    getNewPay(currentSalary, raise, newSalary);

    //display the raise and new salary
    cout << fixed << setprecision(2);
    cout << "Raise: $" << raise << endl;
    cout << "New salary: $" << newSalary << endl;
    return 0;
}  //end of main function
void getRaise(double current, double rate, double &increase)
{
    increase = current * rate;
} //end of getRaise function

void getNewPay(double current, double increase, double &newPay)
{
    newPay = current + increase;
} //end of getNewPay function

LAB 10-5 Desk-Check

The names in black indicate variables that belong to the main function. The names in red indicate variables that belong to the getSquare function. The names in blue indicate variables that belong to the getCube function. The program will display “The sum is: 50.”

<table>
<thead>
<tr>
<th>sum</th>
<th>number</th>
<th>sum</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

50

LAB 10-6 Debug

To debug the program, change the function prototype to either void assignGrade(int pointsEarned, char &letter); or void assignGrade(int, char &); and change the function header to void assignGrade(int pointsEarned, char &letter). Also change the function call to assignGrade(totalPoints, grade);.
Answers to Chapter 11 Mini-Quizzes

Mini-Quiz 11-1

1. a. int quantities[20] = {0};
2. quantities[0]
3. quantities[19]
4. quantities[3] = 7;
5. c. total = getTotal(quantities, 20);

Mini-Quiz 11-2

1. c. total += orders[2];
2. c. if (orders[3] > 25)
3. bonus = sales[0] * 0.15;
4. c. if (sub >= 0 && sub < 10)
5. a. while (x < 20)

Mini-Quiz 11-3

1. if (prices[x] < lowest)
2. sorting
3. for (int x = 0; x < 10; x += 1)
   orders[x] -= 3;
   //end for

Answers to Chapter 11 Labs

LAB 11-1 Stop and Analyze

1. The domestic and international arrays are parallel arrays because the elements in one array are related by their subscripts to the elements in the other array. For example, the first element in both arrays contains the sales made in January; the second element contains the February sales, and so on.


3. The total company sales made in February can be calculated by adding the contents of the domestic[1] element to the contents of the international[1] element.
4. The highest subscript in the international array is 5.

5. The assignment statement would need to be changed to `totalSales += domestic[x – 1] + international[x – 1];`

6. No answer required.

7. //Lab11-1.cpp - calculates the total sales
   //Created/revised by <your name> on <current date>

   #include <iostream>
   using namespace std;

   int main()
   {
       int international[6] = {40000, 75000, 64000, 32600, 47800, 39000};
       int totalDomestic = 0;
       int totalInternational = 0;
       int totalSales = 0;

       for (int x = 0; x < 6; x += 1)
       {
           totalDomestic += domestic[x];
           totalInternational += international[x];
       }  // end for
       totalSales = totalDomestic + totalInternational;

       // display total domestic sales, total international
       // sales, and total sales
       cout << "Total domestic sales: $" << totalDomestic << endl;
       cout << "Total international sales: $" << totalInternational << endl;
       cout << "Total sales: $" << totalSales << endl;
       return 0;
   }  // end of main function

8. //Lab11-1.cpp - calculates the total sales
   //Created/revised by <your name> on <current date>

   #include <iostream>
   using namespace std;

   int main()
   {
int international[6] = {40000, 75000, 64000, 32600, 47800, 39000};

int monthSales[6] = {0};

int totalDomestic = 0;
int totalInternational = 0;
int totalSales = 0;

for (int x = 0; x < 6; x += 1)
{
    totalDomestic += domestic[x];
    totalInternational += international[x];
    monthSales[x] = domestic[x] + international[x];
}  //end for
//display total domestic sales, total international
//sales, and total sales
cout << "Total domestic sales: $" << totalDomestic << endl;
cout << "Total international sales: $" << totalInternational << endl;
cout << "Total sales: $" << totalSales << endl << endl;

//display total sales made in each month
for (int x = 0; x < 6; x += 1)
{
    cout << "Month " << x + 1 << " sales: $" << monthSales[x] << endl;
}  //end for
return 0;
}  //end of main function

LAB 11-2  Plan and Create
No answer required.

LAB 11-3  Modify
//Lab11-3.cpp - stores finish times in an array
//and displays the average and lowest times
//Created/revised by <your name> on <current date>

#include <iostream>
#include <iomanip>
using namespace std;
//function prototypes
void getAverage(double times[], int numElements, double &avg);
void getLowest(double times[], int numElements, double &lowest);

int main()
{
    double finishTimes[5] = {0.0};
    double avgTime = 0.0;
    double lowestTime = 0.0;

    //enter finish times
    for (int x = 0; x < 5; x += 1)
    {
        cout << "Time for race " << x + 1 << ": ";
        cin >> finishTimes[x];
    }  //end for

    getAverage(finishTimes, 5, avgTime);
    getLowest(finishTimes, 5, lowestTime);

    cout << fixed << setprecision(1) << endl;
    cout << "Average 5K finish time: " << avgTime << endl;
    cout << "Lowest 5K finish time: " << lowestTime << endl;
    return 0;
}   //end of main function

//*****function definitions*****
void getAverage(double times[], int numElements, double &avg)
{
    double total = 0.0;

    for (int x = 0; x < numElements; x += 1)
    {
        total += times[x];  //end for
    }
    avg = total / numElements;
}  //end of getAverage function

void getLowest(double times[], int numElements, double &lowest)
{
    lowest = times[0];
    for (int x = 1; x < numElements; x += 1)
    {
        if (times[x] < lowest)
        {
            lowest = times[x];
        }  //end if
    }  //end for
}  //end of getLowest function
LAB 11-4  What's Missing?

//Lab11-4.cpp - Displays the average stock price  
//Created/revised by <your name> on <current date>

#include <iostream>  
#include <iomanip>  
using namespace std;  

//function prototype  
double getTotal(double prices[], int numElements);  

int main()  
{  
    double stockPrices[10] = {35.6, 37.8, 39, 38.9, 38.9,  
                              37.8, 37.8, 37.8, 39, 39.5};  
    double total = 0.0;  
    double average = 0.0;  
    total = getTotal(stockPrices, 10);  
    average = total / 10;  
    cout << fixed << setprecision(2);  
    cout << "Average stock price: $" << average << endl;  
    return 0;  
}  

//*****function definitions*****  
double getTotal(double prices[], int numElements)  
{  
    double sumPrices = 0.0;  
    for (int sub = 0; sub < numElements; sub += 1)  
        sumPrices += prices[sub];  
    //end for  
    return sumPrices;  
}  

//Desk-Check

LAB 11-5  Desk-Check

Desk-check:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>90.0</td>
<td>88.0</td>
<td>77.0</td>
<td>85.0</td>
<td>45.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>100.0</td>
<td>68.0</td>
<td>75.0</td>
<td>85.0</td>
<td>32.0</td>
</tr>
</tbody>
</table>
Answers to Chapter 12 Mini-Quizzes

Mini-Quiz 12-1

1. a. int quantities[6][3] = {0};
2. 18
3. quantities[0][0]
4. quantities[5][2]
5. quantities[1][0] = 20;

The for loop will display the following:
Student 1 average: 95
Student 2 average: 78
Student 3 average: 76
Student 4 average: 85
Student 5 average: 38.5

LAB 11-6 Debug
To debug the program, change the condition in the for clause to x < 10. Also change the increase += quantities[x]; statement in the for loop to quantities[x] += increase;.
### Mini-Quiz 12-2

1. c. total += purchases[2][1];
2. c. if (scores[1][2] > 25)
3. bonus = sales[0][1] * 0.15;
4. a. if (row >= 0 && row < 10)

### Answers to Chapter 12 Labs

#### LAB 12-1  Stop and Analyze

1. 39000
2. The total company sales made in February can be calculated by adding the contents of the `company[0][1]` element to the contents of the `company[1][1]` element.
3. The highest row subscript in the `company` array is 1. The highest column subscript is 5.
4. The January international sales are stored in the `company[1][0]` element.
5. The assignment statement would need to be changed to `companySales += company[location][month - 1];`.
6. No answer required.
7. //Lab12-1.cpp - calculates the total company sales
   //Created/revised by <your name> on <current date>

```cpp
#include <iostream>
using namespace std;

int main()
{
  int company[2][6] = {{75000, 30200, 67800, 45000, 60000, 67500},
                        {40000, 75000, 64000, 32600, 47800, 39000}};

  int companySales = 0;
  int domesticSales = 0;
  int internationalSales = 0;

  for (int month = 0; month < 6; month += 1)
  {
    domesticSales += company[0][month];
    internationalSales += company[1][month];
  }  //end for
  companySales = domesticSales + internationalSales;
```
#include <iostream>
using namespace std;

int main()
{
    int company[2][6] = {{75000, 30200, 67800,
        45000, 60000, 67500},
        {40000, 75000, 64000,
        32600, 47800, 39000}};

    int monthSales[6] = { 0 };
    int companySales = 0;
    int domesticSales = 0;
    int internationalSales = 0;

    for (int month = 0; month < 6; month += 1)
    {
        domesticSales += company[0][month];
        internationalSales += company[1][month];
        monthSales[month] = company[0][month] + company[1][month];
    }  //end for
    companySales = domesticSales + internationalSales;

    cout << "Domestic sales: $" << domesticSales << endl;
    cout << "International sales: $" << internationalSales << endl;
    cout << "Company sales: $" << companySales << endl;

    for (int x = 0; x < 6; x += 1)
    {
        cout << "Month " << x + 1 << " sales: $" << monthSales[x] << endl;
    }  //end for
    return 0;
}   //end of main function
**LAB 12-3  Modify**

//Lab12-3.cpp - displays the shipping charge  
//Created/revised by <your name> on <current date>

#include <iostream>
using namespace std;

int main()
{
    int shipCharges[3][2] = {{101, 0},
                              {51, 10},
                              {1, 20}};
    int numOrdered = 0;
    int rowSub = 0;
    
    cout << "Number ordered " <<
    "(negative number or 0 to end): ";
    cin >> numOrdered;
    
    while (numOrdered > 0 && numOrdered <= 999999)
    {
        //search array
        rowSub = 0;
        while (rowSub < 3 && numOrdered < shipCharges[rowSub][0])
            rowSub += 1;
        //end while
        
        cout << "Shipping charge for a quantity of " <<
        " is $" << numOrdered << " " << shipCharges[rowSub][1] << endl;
        
        cout << "Number ordered " <<
        "(negative number or 0 to end): ";
        cin >> numOrdered;
    }  //end while
    return 0;
}  //end of main function

**LAB 12-4  What's Missing?**

//Lab12-4.cpp - Displays the average stock price  
//Created/revised by <your name> on <current date>

#include <iostream>
#include <iomanip>
using namespace std;

//function prototype
double getTotal(double prices[2][5]);
int main()
{
    double stockPrices[2][5] = {{35.6, 37.8, 39, 38.9, 38.9},
                                {37.8, 37.8, 37.8, 39, 39.5}};
    double total = 0.0;
    double average = 0.0;

    total = getTotal(stockPrices);
    average = total / 10;
    cout << fixed << setprecision(2);
    cout << "Average stock price: $" << average << endl;
    return 0;
} //end of main function

//*****function definitions*****
double getTotal(double prices[2][5])
{
    double sumPrices = 0.0;
    for (int row = 0; row < 2; row += 1)
        for (int col = 0; col < 5; col += 1)
            sumPrices += prices[row][col];
    return sumPrices;
} //end of getTotal function

---

**LAB 12-5 Desk-Check**

<table>
<thead>
<tr>
<th>sales[0][0]</th>
<th>sales[0][1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3567.85</td>
<td>2589.99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sales[1][0]</th>
<th>sales[1][1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3239.67</td>
<td>2785.55</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sales[2][0]</th>
<th>sales[2][1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1530.5</td>
<td>1445.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>total</th>
<th>store</th>
<th>book</th>
</tr>
</thead>
<tbody>
<tr>
<td>12183.06</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>13713.56</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>15159.36</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
### Answers to Mini-Quizzes and Labs

**Answers to Mini-Quizzes**

**Mini-Quiz 13-1**

1. c. `const string CITY = "Fort Knox";`
2. b. `string state = "";`
3. a. `getline(cin, streetAddress, '\n');`
4. c. `cin.ignore(10, '\n');`

**Mini-Quiz 13-2**

1. a. `while (employee.length() > 20)`
2. if `code.length() == 7)`
3. d. both a and b
4. `cout << college.substr(college.length() - 1);`

**Mini-Quiz 13-3**

1. a. `location = cityState.find("", 0);`
2. 13
3. d. all of the above

**Mini-Quiz 13-4**

1. d. none of the above
2. a. `sentence = sentence + temp.assign(4, '!' );`
3. b. `areaCode = "(" + areaCode + ")";`
Answers to Chapter 13 Labs

LAB 13-1 Stop and Analyze

1. The purpose of the loop on Lines 20 through 33 is to access each character in the sales variable, one character at a time, in order to count the number of numbers, periods, and other characters.

2. The statement on Line 22 assigns the current character from the sales variable to the currentChar variable.

3. The selection structure on Lines 35 through 39 determines whether the sales amount entered by the user is either valid or invalid. An invalid sales amount is one that either contains more than one period or contains at least one character that is either not a number and not a period.

4. The statement on Line 32 allows the program to access the next character in the sales variable.

5. No answer required.

6. //Lab13-1.cpp
   //Created/revised by <your name> on <current date>

   #include <iostream>
   #include <string>
   using namespace std;

   int main()
   {
     string sales = "";
     string currentChar = "";
     string otherChars = "";
     int numNumbers = 0;
     int numPeriods = 0;
     int numOtherChars = 0;

     cout << "Sales amount: ";
     getline(cin, sales);

     for (int sub = 0; sub < sales.length(); sub += 1)
     {
       currentChar = sales.substr(sub, 1);
       if (currentChar == ".")
         numPeriods += 1;
       else
         if (currentChar < "0" || currentChar > "9")
           numOtherChars += 1;
       otherChars += currentChar;
     }
   }
```cpp
numNumbers += 1;
// end if
// end if
} // end for
if (numPeriods > 1 || numOtherChars > 0)
  cout << "Invalid sales amount" << endl;
else
  cout << "Valid sales amount" << endl;
// end if
cout << "Numbers: " << numNumbers << endl;
cout << "Periods: " << numPeriods << endl;
cout << numOtherChars << " other characters: "
  << otherChars << endl;
return 0;
} // end of main function
```

---

**LAB 13-2  Plan and Create**

No answer required.

---

**LAB 13-3  Modify**

//Lab13-3.cpp - Guess the Word game
//Created/revised by <your name> on <current date>

```cpp
#include <iostream>
#include <string>
// #include <cstdlib>
using namespace std;

int main()
{
  string origWord = "";
  string letter = "";
  char dashReplaced = 'N';
  char gameOver = 'N';
  int numIncorrect = 0;
  string displayWord = "-----";
  int numChars = 0;

  // get original word
  cout << "Enter a word in uppercase: ";
  getline(cin, origWord);

```
numChars = origWord.length();
displayWord.assign(numChars, '-');

system("cls"); //clear the screen

//start guessing
cout << "Guess this word: " << displayWord << endl;
while (gameOver == 'N')
{
    cout << "Enter an uppercase letter: "; cin >> letter;

    //search for the letter in the original word
    for (int x = 0; x < numChars; x += 1)
    {
        //if the current character matches
        //the letter, replace the corresponding
        //dash in the displayWord variable and then
        //set the dashReplaced variable to 'Y'
        if (origWord.substr(x, 1) == letter)
        {
            displayWord.replace(x, 1, letter);
            dashReplaced = 'Y';
        }  //end if
    }  //end for

    //if a dash was replaced, check whether the
    //displayWord variable contains another dash
    if (dashReplaced == 'Y')
    {
        //if the displayWord variable does not
        //contain any dashes, the game is over
        if (displayWord.find("-", 0) == -1)
        {
            gameOver = 'Y';
            cout << endl << "Yes, the word is " << origWord << endl;
            cout << "Great guessing!" << endl;
        }  //end if
    }  else  //otherwise, continue guessing
    {
        cout << endl << "Guess this word: " << displayWord << endl;
        dashReplaced = 'N';
    }  //end if
}  else  //processed when dashReplaced contains 'N'
{
    //add 1 to the number of incorrect guesses
    numIncorrect += 1;
    //if the number of incorrect guesses is 10,
//the game is over
if (numIncorrect == 10)
{
    gameOver = 'Y';
    cout << endl << "Sorry, the word is "
         << origWord << endl;
} //end if
} //end if
} //end while
return 0;
} //end of main function

LAB 13-4 What's Missing?

//Lab13-4.cpp - displays a comma (if necessary) in the output
//Created/revised by <your name> on <current date>

#include <iostream>
#include <string>
using namespace std;

int main()
{
    string population = "";
    int highSub = 0;

    cout << "Enter the population: ";
    cin >> population;
    highSub = population.length() - 1;

    while (highSub >= 3)
    {
        population.insert(highSub - 2, ",");
        highSub -= 3;
    } //end while

    cout << population << endl;
    return 0;
} //end of main function

LAB 13-5 Desk-Check

Desk-check:

message subMessage1 subMessage2
vexpr
tj
vexpr
Answers to Chapter 14 Mini-Quizzes

Mini-Quiz 14-1
1. b. #include <fstream>
2. d. ios::in
3. the Boolean value false
4. ofstream outAlbums;
5. c. outAlbums.open("mine.txt", ios::app);

Mini-Quiz 14-2
1. c. outInv << quantity << endl;
2. d. outFile << score1 << '#' << score2 << endl;
3. d. inInv >> number;

Mini-Quiz 14-3
1. c. while (!inInv.eof())
2. the Boolean value false
3. outInv.close();
Answers to Chapter 14 Labs

**LAB 14-1  Stop and Analyze**

1. The instruction on Line 5 is necessary because the program uses the `cin` and `cout` objects. The instruction on Line 6 is necessary because the program uses the `string` class. The instruction on Line 7 is necessary because the program uses the `ofstream` class.

2. Each record contains two fields: the movie title and the year the movie was released.

3. The `movies.txt` file will contain only the two records written during the second run of the program. This is because the program opens the `movies.txt` file for output, which means the file's contents will be erased each time the program is run.

4. To save the previous records, you need to open the file for append. You do this by changing the `mode` in the `open` function in Line 17 to `ios::app`.

5. `if (outFile.is_open() == true)`

6. The purpose of the `#` character is to separate the movie title field from the year released field in each record.

7. The statement in Line 32 closes the output file. Neglecting to close a file can result in a loss of data.

8. No answer required.

9. No answer required.

10. No answer required.

11. Change the `outFile.open("movies.txt", ios::out);` statement in Line 17 to `outFile.open("movies.txt", ios::app);`.

12. No answer required.

**LAB 14-2  Plan and Create**

No answer required.

**LAB 14-3  Modify**

//Lab14-3.cpp - saves records to a sequential access file and also calculates and displays the total of the sales amounts stored in the file
//Displays the records and the average sales amount
//Created/revised by <your name> on <current date>
#include <iostream>
#include <string>
#include <fstream>
using namespace std;

//function prototypes
int getChoice();
void addRecords();
void displayRecords();
void displayTotal();
void displayAvg();

int main()
{
    int choice = 0;
do
    {
        //get user's menu choice
        choice = getChoice();
        if (choice == 1)
            addRecords();
        else if (choice == 2)
            displayRecords();
        else if (choice == 3)
            displayTotal();
        else if (choice == 4)
            displayAvg();
        //end if
    }    while (choice != 5);
    return 0;
}    //end of main function

//*****function definitions*****
int getChoice()
{
    //displays menu and returns choice
    int menuChoice = 0;
    cout << endl << "Menu Options" << endl;
    cout << "1  Add Records" << endl;
    cout << "2  Display Records" << endl;
    cout << "3  Display Total Sales" << endl;
    cout << "4  Display Average Sales" << endl;
    cout << "5  Exit" << endl;
    cout << "Choice (1 through 5)? ";
    cin >> menuChoice;
    cin.ignore(100, '\n');
    cout << endl;
    return menuChoice;
}    //end of getChoice function
void addRecords()
{
    //saves records to a sequential access file
    string name = "";
    int sales = 0;
    ofstream outFile;

    //open file for append
    outFile.open("sales.txt", ios::app);

    //if the open was successful, get the
    //salesperson's name and sales amount and
    //then write the information to the file;
    //otherwise, display an error message
    if (outFile.is_open())
    {
        cout << "Salesperson's name (X to stop): ";
        getline(cin, name);
        while (name != "X" && name != "x")
        {
            cout << "Sales: ";
            cin >> sales;
            cin.ignore(100, '\n');

            outFile << name << '#' << sales << endl;

            cout << "Salesperson's name " << "(X to stop): ";
            getline(cin, name);
        } //end while
    outFile.close();
    }
else
    cout << "sales.txt file could not be opened" << endl;
} //end if
} //end of addRecords function

void displayRecords()
{
//displays the contents of the sales.txt file
    string name = "";
    int sales = 0;
    ifstream inFile;

    //open file for input
    inFile.open("sales.txt");

    //if the open was successful, read a
    //record and then display the record
    //otherwise, display an error message
    if (inFile.is_open())
void displayRecords()
{
    ifstream inFile;
    inFile.open("sales.txt");

    if (inFile.is_open())
    {
        string name = "";
        int sales = 0;
        int total = 0;

        while (!inFile.eof())
        {
            getline(inFile, name, '#');
            inFile >> sales;
            inFile.ignore();

            cout << name << "$" << sales << endl;
            getline(inFile, name, '#');
            inFile >> sales;
            inFile.ignore();
        } //end while

        inFile.close();
    } //end if

} //end of displayRecords function

void displayTotal()
{
    ifstream inFile;
    inFile.open("sales.txt");

    if (inFile.is_open())
    {
        string name = "";
        int sales = 0;
        int total = 0;

        while (!inFile.eof())
        {
            total += sales;
            getline(inFile, name, '#');
            inFile >> sales;
            inFile.ignore();
        } //end while

        inFile.close();
        cout << "Total sales $" << total << endl;
    } //end if

} //end of displayTotal function
} 
else  
    cout << "sales.txt file could not be opened"  
    << endl;  
//end if  
}  //end of displayTotal function

void displayAvg()
{
    //calculates and displays the average sales 
    string name = ""; 
    int sales = 0; 
    int totalSales = 0; 
    int numSales = 0; 
    double avgSales = 0.0; 
    ifstream inFile; 

    //open file for input 
    inFile.open("sales.txt"); 

    //if the open was successful, read the 
    //salesperson's name and sales amount, then add 
    //the sales amount to the accumulator and add 1 
    //to the counter; otherwise, display an error message 
    if (inFile.is_open())
    {
        getline(inFile, name, '#'); 
        inFile >> sales; 
        inFile.ignore(); 

        while (!inFile.eof())
        {
            totalSales += sales; 
            numSales += 1; 
            getline(inFile, name, '#'); 
            inFile >> sales; 
            inFile.ignore(); 
        } //end while 
    inFile.close(); 

    //calculate and display the average sales 
    avgSales = 
        static_cast<double>(totalSales) / numSales; 
    cout << "Average sales $" << avgSales 
        << endl << endl;  
    }
else  
    cout << "sales.txt file could not be opened."  
    << endl;  
//end if  
}  //end of displayAvg function
//Lab14-4.cpp - saves records to and reads records from a file
//Created/revised by <your name> on <current date>

#include <iostream>
#include <string>
#include <fstream>
using namespace std;

int main()
{
    string firstName = "";
    string lastName = "";
    ifstream inFile;
    ofstream outFile;

    outFile.open("Lab14-4.txt", ios::out);
    for (int num = 1; num < 6; num += 1)
    {
        cout << "First name: ";
        cin >> firstName;
        cout << "Last name: ";
        cin >> lastName;
        outFile << firstName << '#' << lastName << endl;
    }  //end for
    outFile.close();

    inFile.open("Lab14-4.txt", ios::in);
    if (inFile.is_open())
    {
        for (int num = 1; num < 6; num += 1)
        {
            getline(inFile, firstName, '#');
            getline(inFile, lastName, '\n');
            cout << lastName << ", " << firstName << endl;
        }  //end for
        inFile.close();
    }
    else
    
        cout << "Can't locate the Lab14-4.txt file."
     //end if
    return 0;
}    //end of main function
**Lab 14-5 Desk-Check**

Desk-check:

<table>
<thead>
<tr>
<th>store1Sales</th>
<th>store2Sales</th>
<th>store1Total</th>
<th>store2Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15400</td>
<td>23600</td>
<td>15400</td>
<td>23600</td>
</tr>
<tr>
<td>12345</td>
<td>25350</td>
<td>27745</td>
<td>48950</td>
</tr>
<tr>
<td>16750</td>
<td>31000</td>
<td>48455</td>
<td>80150</td>
</tr>
</tbody>
</table>

The code will display the following:

- Store 1's total sales: $62945
- Store 2's total sales: $110850

**Lab 14-6 Debug**

To debug the program, add the `cin.ignore(100, 'n');` statement below the `cin >> num2;` statement.

---

**Answers to Chapter 15 Mini-Quizzes**

**Mini-Quiz 15-1**

1. object-oriented programming
2. b. False
3. b. an instance of the class
4. b. behaviors

**Mini-Quiz 15-2**

1. b. False
2. declaration, implementation
3. d. variable declarations
4. a. True
5. Check `payCheck;`
6. a. `paycheck.getCheck()`
Mini-Quiz 15-3

1. scope resolution operator
2. Item();
3. Item::Item()
   {
       code = ' ';
       price = 0;
   } //end of default constructor

Mini-Quiz 15-4

1. signature
2. Item(int);
3. overloaded

Answers to Chapter 15 Labs

**LAB 15-1  Stop and Analyze**

1. The names of the private data members are id and price.
2. The name of the default constructor is Item. The constructor’s purpose is to initialize the private variables in the class.
3. The setData method assigns the program values to the private variables in the class.
4. The getIncreasedPrice method calculates and returns the new price of the item.
5. The code on Line 36 determines whether the value stored in the rate variable is greater than 1.0, which indicates that the rate was entered as an integer rather than as a decimal number. (For example, the user entered 15 rather than .15 as the rate.) The code on Line 37 converts the integer to its decimal equivalent by dividing the integer by 100.
6. The computer.setData(computerId, computerPrice); statement is missing from Line 69.
7. The << computer.getIncreasedPrice(incRate) code is missing from Line 74.
LAB 15-2 Plan and Create

No answer required.

LAB 15-3 Modify

In the Lab15-3.cpp file, change double lawnLength = 0.0; to int lawnLength = 0; Also change double lawnWidth = 0.0; to int lawnWidth = 0.; In the Lab15-3 Rectangle.h file, add the void setDimensions(int, int); prototype below the existing setDimensions prototype. Also add the following setDimensions function:

```cpp
void Rectangle::setDimensions(int len, int wid)
{
    //assigns dimensions to private data members
    if (len > 0 && wid > 0)
    {
        length = len;
        width = wid;
    }  //end if
} //end of setDimensions method
```

LAB 15-4 What's Missing?

//Parallelogram.h
//Created/revised by <your name> on <current date>

//declaration section
class Parallelogram
{
    public:
        Parallelogram();
        double calcArea(double, double);
    private:
        double length;
        double height;
};

//implementation section
Parallelogram::Parallelogram()
{
    length = 0.0;
    height = 0.0;
}  //end of default constructor
double Parallelogram::calcArea(double l, double h)
{
    double area = 0.0;

    if (l >= 0.0 && h >= 0.0)
        area = l * h;
    else
        area = -1;
    //end if

    return area;
}  //end of calcArea method

//Lab15-4.cpp - displays the area of a parallelogram in square yards
//Created/revised by <your name> on <current date>

#include <iostream>
#include <iomanip>
#include "Parallelogram.h"
using namespace std;

int main()
{
    Parallelogram parkingLot;
    double lotLength = 0.0;
    double lotHeight = 0.0;
    double lotArea = 0.0;

    cout << "Length (in feet): ";
    cin >> lotLength;
    cout << "Height (in feet): ";
    cin >> lotHeight;

    lotArea = parkingLot.calcArea(lotLength, lotHeight) / 9;
    cout << fixed << setprecision(2) << endl;
    cout << "Square yards: " << lotArea << endl;
    return 0;
}  //end of main function


LAB 15-5  Desk-Check

Desk-check:

<table>
<thead>
<tr>
<th>BONUS_RATE</th>
<th>name</th>
<th>dollars</th>
<th>bonus</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>Carla Rensen</td>
<td>12456.75</td>
<td>622.84</td>
</tr>
</tbody>
</table>
The code will display the following: Carla Rensen bonus: $622.84

LAB 15-6 Debug
To debug the program, enter the `#include "Lab15-6 Inventory.h"` directive in the Lab15-6.cpp file. Also change the `if` clause in the `setItem` method in the Lab15-6 Inventory.h file to `if (num > 0)`.