CHAPTER 4

TYPE CONVERSIONS

The answers for the Type Conversions section are located at the end of the section.

Examine the first eight expressions shown here. If the expression requires an implicit type conversion, explain how the expression will be evaluated; use Figures 4-9 and 4-10 in the book as a guide. In the expressions, `quantity` is an `int` variable, `sales` is a `double` variable, and `TAX_RATE` is a `double` named constant. The `quantity` and `sales` variables contain the numbers 10 and 500.0, respectively. The `TAX_RATE` named constant contains the number .05.

1. `100 * 1.5`
2. `sales / 2`
3. `sales * TAX_RATE`
4. `quantity * 3 * TAX_RATE`
5. `quantity / 2.0`
6. `quantity + 15`
7. `sales / quantity`
8. `static_cast<double>(quantity) / 2`

9. A student earns a total of 353 points on five tests. The total points are stored in an `int` variable named `totalPoints`. Will the `totalPoints / 5` expression calculate the correct average test score? If not, modify the expression so that it will.

ANSWERS FOR THE TYPE CONVERSIONS SECTION

1. The integer 100 is implicitly promoted to the `double` number 100.0 before being multiplied by the `double` number 1.5. The result is the `double` number 150.0.
2. The integer 2 is implicitly promoted to the `double` number 2.0 before being divided into the `double` number 500.0. The result is the `double` number 250.0.
3. This expression does not require any implicit type conversion.
4. The integer 10 is multiplied by the integer 3, giving 30. The integer 30 is then implicitly promoted to the `double` number 30.0 before being multiplied by the `double` number .05. The result is the `double` number 1.5.
5. The integer 10 is implicitly promoted to the `double` number 10.0 before being divided by the `double` number 2.0. The result is the `double` number 5.0.
6. This expression does not require any implicit type conversion.
7. The integer 10 is implicitly promoted to the `double` number 10.0 before being divided into the `double` number 500.0. The result is the `double` number 50.0.
8. The integer 10 is explicitly promoted to the `double` number 10.0. The integer 2 is then implicitly promoted to the `double` number 2.0 before being divided into the `double` number 10.0. The result is the `double` number 5.0.

9. The `totalPoints / 5` expression will not calculate the correct average test score. You can use any of the following expressions to calculate the correct average test score.

```
totalPoints / 5.0
static_cast<double>(totalPoints) / 5
static_cast<double>(totalPoints) / 5.0
static_cast<float>(totalPoints) / 5
```

ASSIGNMENT STATEMENTS

The answers for the Assignment Statements section are located at the end of the section.

1. Write an assignment statement that assigns the integer 2500 to an `int` variable named `population`.
2. Write an assignment statement that assigns the sum of two `double` variables named `sales1` and `sales2` to a `double` variable named `totalSales`.
3. Write an assignment statement that divides the integer 7 by the integer 3 and then assigns the result to a `double` variable named `answer`.
4. Write an assignment statement that assigns the letter X to a `char` variable named `letter`.
5. Write an assignment statement that assigns the string "Louisville, KY" to a `string` variable named `cityState`.
6. Write an assignment statement that multiplies the contents of a `double` variable named `sales` by the contents of the `double BONUS_RATE` named constant, and then assigns the result to a `double` variable named `bonus`.
7. Write an assignment statement that increases the contents of a `double` variable named `sales` by 2%.

ANSWERS FOR THE ASSIGNMENT STATEMENTS SECTION

1.
```
population = 2500;
```
2.
```
totalSales = sales1 + sales2;
```
3. You can use any of the following:
```
answer = 7.0 / 3.0;
answer = 7 / 3.0;
answer = 7.0 / 3;
answer = static<double>(7) / static<double>(3);
answer = static<double>(7) / 3;
answer = 7 / static<double>(3);
answer = static<double>(7) / 3.0;
answer = 7.0 / static<double>(3);
```
4.
```
letter = 'X';
```
5.
```
cityState = "Louisville, KY";
```
6.
```
bonus = sales * BONUS_RATE;
```
7. `sales = sales * 1.02;` (or you can use `sales = sales + sales * .02;`)

CODING ALGORITHMS

Code the following 10 algorithms. The answers for the Coding Algorithms section are located at the end of the section.

1.

IPO chart information	C++ instructions
Input length width **Processing** none **Output** area **Algorithm** 1. enter the length and width 2. calculate the area by multiplying the length by the width 3. display the area	

WM-Figure 4-1 IPO chart for the Quality Builders problem

2.

IPO chart information	C++ instructions
Input current price increase percentage **Processing** none **Output** increase amount new price **Algorithm** 1. enter the current price and increase percentage 2. calculate the increase amount by multiplying the current price by the increase percentage 3. calculate the new price by adding the increase amount to the current price 4. display the increase amount and new price	

WM-Figure 4-2 IPO chart for the Toys Are Fun problem

3.

IPO chart information	C++ instructions
Input current annual salary bonus percentage	
Processing none	
Output bonus amount	
Algorithm 1. enter the current annual salary and bonus percentage 2. calculate the bonus amount by multiplying the current annual salary by the bonus percentage 3. display the bonus amount	

WM-Figure 4-3 IPO chart for the Dellso Incorporated problem

4. NOTE: The input items in the IPO chart are integers.

IPO chart information	C++ instructions
Input first number second number	
Processing none	
Output average	
Algorithm 1. enter the first number and second number 2. calculate the average by adding the first number to the second number, and then dividing the sum by 2 3. display the average	

WM-Figure 4-4 IPO chart for the Mary Hernandez problem

5.

IPO chart information	C++ instructions
Input	
beginning inventory	
amount sold	
amount returned	
Processing	
none	
Output	
ending inventory	
Algorithm	
1. enter the beginning inventory, amount sold, and amount returned	
2. calculate the ending inventory by subtracting the amount sold from the beginning inventory, and then adding the amount returned to the result	
3. display the ending inventory	

WM-Figure 4-5 IPO chart for the Universal Heating and Cooling problem

6.

IPO chart information	C++ instructions
Input	
property tax rate	
assessed value	
Processing	
none	
Output	
annual property tax	
Algorithm	
1. enter the property tax rate and assessed value	
2. calculate the annual property tax by dividing the assessed value by 100, and then multiplying the result by the property tax rate	
3. display the annual property tax	

WM-Figure 4-6 IPO chart for the city of Joliet problem

7.

IPO chart information **C++ instructions**

Input

number of envelopes
number of pages
envelope charge
page charge

Processing

none

Output

amount due for envelopes
amount due for pages
total due

Algorithm

1. enter the number of envelopes, number of pages, envelope charge, and page charge
2. calculate the amount due for envelopes by multiplying the number of envelopes by the envelope charge
3. calculate the amount due for pages by multiplying the number of pages by the page charge
4. calculate the total due by adding the amount due for envelopes to the amount due for pages
5. display the amount due for envelopes, the amount due for pages, and the total due

WM-Figure 4-7 IPO chart for the Typing Haven problem

8.

IPO chart information **C++ instructions**

Input

diameter
price per foot
pi (3.14)

Processing

none

Output

circumference
total price

Algorithm

1. enter the diameter and price per foot
2. calculate the circumference by multiplying the diameter by pi
3. calculate the total price by multiplying the circumference by the price per foot
4. display the circumference and total price

WM-Figure 4-8 IPO chart for the Builders Inc. problem

9.

IPO chart information	C++ instructions
Input length in feet width in feet square foot price **Processing** none **Output** area total price **Algorithm** 1. enter the length in feet, width in feet, and square foot price 2. calculate the area by multiplying the length in feet by the width in feet 3. calculate the total price by multiplying the area by the square foot price 4. display the area and total price	

WM-Figure 4-9 IPO chart for the Everyday Tile problem

10.

IPO chart information	C++ instructions
Input gross pay tax deduction insurance deduction **Processing** none **Output** net pay **Algorithm** 1. enter the gross pay 2. calculate the net pay by subtracting the tax deduction and insurance deduction from the gross pay 3. display the net pay	

WM-Figure 4-10 IPO chart for the Johnson Industries problem

ANSWERS FOR THE CODING ALGORITHMS SECTION

1.

IPO chart information	C++ instructions
Input	
length	`double length = 0.0;`
width	`double width = 0.0;`
Processing	
none	
Output	
area	`double area = 0.0;`
Algorithm	

1. enter the length and width	`cout << "Length: ";`
	`cin >> length;`
	`cout << "Width: ";`
	`cin >> width;`

2. calculate the area by multiplying the length by the width	`area = length * width;`
3. display the area	`cout << "Area: " << area << endl;`

WM-Figure 4-11 IPO chart for the Quality Builders problem

2.

IPO chart information	C++ instructions
Input	
current price	`double curPrice = 0.0;`
increase percentage (15%)	`const double INCREASE_RATE = .15;`
Processing	
none	
Output	
increase amount	`double increase = 0.0;`
new price	`double newPrice = 0.0;`
Algorithm	
1. enter the current price	`cout << "Current price: ";` `cin >> curPrice;`
2. calculate the increase amount by multiplying the current price by the increase percentage	`increase = curPrice * INCREASE_RATE;`
3. calculate the new price by adding the increase amount to the current price	`newPrice = curPrice + increase;`
4. display the increase amount and new price	`cout << "Increase: " << increase << endl;` `cout << "New price: " << newPrice << endl;`

WM-Figure 4-12 IPO chart for the Toys Are Fun problem

3.

IPO chart information	C++ instructions
Input	
current annual salary bonus percentage	`double curSalary = 0.0;` `double bonusRate = 0.0;`
Processing	
none	
Output	
bonus amount	`double bonus = 0.0;`
Algorithm	

1. enter the current annual salary and bonus percentage	`cout << "Current annual salary: ";`
	`cin >> curSalary;`
	`cout << "Bonus rate (in decimal form): ";`
	`cin >> bonusRate;`

2. calculate the bonus amount by multiplying the current annual salary by the bonus percentage	`bonus = curSalary * bonusRate;`
3. display the bonus amount	`cout << "Bonus: " << bonus << endl;`

WM-Figure 4-13 IPO chart for the Dellso Incorporated problem

4.

IPO chart information	C++ instructions
Input	
first number second number	`int num1 = 0;` `int num2 = 0;`
Processing	
none	
Output	
average	`double avg = 0.0;`
Algorithm	

1. enter the first number and second number	`cout << "First number: ";`
	`cin >> num1;`
	`cout << "Second number: ";`
	`cin >> num1;`

2. calculate the average by adding the first number to the second number, and then dividing the sum by 2	`avg = (num1 + num2) / 2.0;`
3. display the average	`cout << "Average: " << avg << endl;`

WM-Figure 4-14 IPO chart for the Mary Hernandez problem

5.

IPO chart information	C++ instructions
Input	

beginning inventory
amount sold
amount returned
	`int beginInv = 0;`
	`int sold = 0;`
	`int returned = 0;`

Processing none	
Output ending inventory	`int endInv = 0;`

Algorithm
1. enter the beginning inventory, amount sold, and amount returned
	`cout << "Beginning inventory: ";`
	`cin >> beginInv;`
	`cout << "Sold: ";`
	`cin >> sold;`
	`cout << "Returned: ";`
	`cin >> returned;`

2. calculate the ending inventory by subtracting the amount sold from the beginning inventory, and then adding the amount returned to the result	`endInv = beginInv - sold + returned;`
3. display the ending inventory	`cout << "Ending inventory: " << endInv << endl;`

WM-Figure 4-15 IPO chart for the Universal Heating and Cooling problem

6.

IPO chart information	C++ instructions
Input	
property tax rate assessed value	`double taxRate = 0.0;` `int assessedValue = 0;`
Processing none	
Output annual property tax	`double tax = 0;`

Algorithm
1. enter the property tax rate and assessed value
	`cout << "Tax rate (in decimal form): ";`
	`cin >> taxRate;`
	`cout << "Assessed value: ";`
	`cin >> assessedValue;`

2. calculate the annual property tax by dividing the assessed value by 100, and then multiplying the result by the property tax rate	`tax = assessedValue / 100 * taxRate;`
3. display the annual property tax	`cout << "Annual property tax: " << tax << endl;`

WM-Figure 4-16 IPO chart for the city of Joliet problem

7.

IPO chart information	C++ instructions
Input	
number of envelopes	`int envelopes = 0;`
number of pages	`int pages = 0;`
envelope charge	`double envelopeChg = 0.0;`
page charge	`double pageChg = 0.0;`
Processing	
none	
Output	
amount due for envelopes	`double dueEnvelopes = 0.0;`
amount due for pages	`double duePages = 0.0;`
total due	`double dueTotal = 0.0;`
Algorithm	

1. enter the number of envelopes, number of pages, envelope charge, and page charge	`cout << "Number of envelopes: ";`
	`cin >> envelopes;`
	`cout << "Number of pages: ";`
	`cin >> pages;`
	`cout << "Envelope charge: ";`
	`cin >> envelopeChg;`
	`cout << "Page charge: ";`
	`cin >> pageChg;`

2. calculate the amount due for envelopes by multiplying the number of envelopes by the envelope charge	`dueEnvelopes = envelopes * envelopeChg;`
3. calculate the amount due for pages by multiplying the number of pages by the page charge	`duePages = pages * pageChg;`
4. calculate the total due by adding the amount due for envelopes to the amount due for pages	`dueTotal = dueEnvelopes + duePages;`

5. display the amount due for envelopes, the amount due for pages, and the total due	`cout << "Due for envelopes: " << dueEnvelopes << endl;`
	`cout << "Due for pages: " << duePages << endl;`
	`cout << "Total due: " << dueTotal << endl;`

WM-Figure 4-17 IPO chart for the Typing Haven problem

8.

IPO chart information	C++ instructions
Input	
diameter	`double diameter = 0.0;`
price per foot	`double pricePerFt = 0.0;`
pi (3.14)	`const double PI = 3.14;`
Processing	
none	
Output	
circumference	`double circumference = 0.0;`
total price	`double totalPrice = 0.0;`

Algorithm
1. enter the diameter and price per foot
	`cout << "Circle diameter: ";`
	`cin >> diameter;`
	`cout << "Price per foot: ";`
	`cin >> pricePerFt;`

2. calculate the circumference by multiplying the diameter by pi	`circumference = diameter * PI;`

3. calculate the total price by multiplying the circumference by the price per foot
4. display the circumference and total price
	`totalPrice = circumference * pricePerFt;`
	`cout << "Circumference: " << circumference << endl;`
	`cout << "Total price: " << totalPrice << endl;`

WM-Figure 4-18 IPO chart for the Builders Inc. problem

9.

IPO chart information	**C++ instructions**

Input
length in feet
width in feet
square foot price
	`double length = 0.0;`
	`double width = 0.0;`
	`double priceSqFt = 0.0;`

Processing none	
Output area total price	`double area = 0.0;` `double totalPrice = 0.0;`

Algorithm
1. enter the length in feet, width in feet, and square foot price
	`cout << "Length (feet): ";`
	`cin >> length;`
	`cout << "Width (feet): ";`
	`cin >> width;`
	`cout << "Price per square foot: ";`
	`cin >> priceSqFt;`

2. calculate the area by multiplying the length in feet by the width in feet	`area = length * width;`

3. calculate the total price by multiplying the area by the square foot price
4. display the area and total price
	`totalPrice = area * priceSqFt;`
	`cout << "Area: " << area << endl;`
	`cout << "Total price: " << totalPrice << endl;`

WM-Figure 4-19 IPO chart for the Everyday Tile problem

10.

IPO chart information	C++ instructions
Input	
gross pay	`double gross = 0.0;`
tax deduction	`double tax = 0.0;`
insurance deduction	`double insurance = 0.0;`
Processing	
none	
Output	`double netPay = 0.0;`
net pay	
Algorithm	`cout << "Gross pay: ";`
1. enter the gross pay	`cin >> gross;`
2. calculate the net pay by subtracting the tax deduction and insurance deduction from the gross pay	`netPay = gross - tax - insurance;`
3. display the net pay	`cout << "Net pay: " << netPay << endl;`

WM-Figure 4-20 IPO chart for the Johnson Industries problem