

# SOLUTIONS MANUAL

SIXTH EDITION

Starting Out with

C++

*Early Objects*

Tony Gaddis  
Judy Walters  
Godfrey Muganda



21. *Input weeks* // with prompt  
*days = weeks \* 7*  
*Display days*
22. *Input eggs* // with prompt  
*cartons = eggs / 12* // perform integer divide  
*Display cartons*
23. *Input speed* // with prompt  
*Input time* // with prompt  
*distance = speed \* time*  
*Display distance*
24. *Input miles* // with prompt  
*Input gallons* // with prompt  
*milesPerGallon = miles / gallons*  
*Display milesPerGallon*
25. A) 0  
100  
B) 8  
2  
C) I am the incredible computing machine  
and I will amaze you.
26. A) Be careful!  
This might/n be a trick question.  
B) 23  
1
27. The C-style comments symbols are backwards.  
*iostream* should be enclosed in angle brackets.  
There shouldn't be a semicolon after *int main()*.  
The opening and closing braces of function *main* are reversed.  
There should be a semicolon after *int a, b, c*.  
The comment `\\ Three integers` should read `// Three integers`.  
There should be a semicolon at the end of each of the three assignment statements.  
*cout* begins with a capital letter.  
The stream insertion operator (that appears twice in the *cout* statement)  
should read `<<` instead of `<`.  
The *cout* statement uses the variable *C* instead of *c*.

28. Whatever problem a pair of students decides to work with they must determine such things as which values will be input vs. which will be set internally in the program, how much precision is required on calculations, what output will be produced by the program, and how it should be displayed. Students must also determine how to handle situations that are not clear cut. In the paint problem many of these considerations are listed in the teacher answer key (Chapter 1, Question 34). In the recipe program students must determine such things as how to handle quantities, like one egg, that cannot be halved. In the driving program, knowing distance and speed are not enough. Agreement should be reached on how to handle delays due to traffic lights and traffic congestion. Should this be an input value, computed as a percent of overall driving time, or handled some other way?

### Chapter 3

1. A) `cin >> description;`  
B) `getline(cin, description);`
2. `char name[35];`
3. A) `cin >> setw(25) >> name;`  
B) `cin.getline(name, 25);`
4. `cin >> age >> pay >> section;`
5. `iostream` and `iomanip`
6. `char city[31];`
7. A) `price = 12 * unitCost;`  
B) `cout << setw(12) << 98.7;`  
C) `cout << 12;`
8. 5, 22, 20, 6, 46, 30, 0, 3, 16
9. A) `a = 12 * x;`  
B) `z = 5 * x + 14 * y + 6 * k;`  
C) `y = pow(x, 4);`  
D) `g = (h + 12) / (4 * k);`  
E) `c = pow(a, 3) / (pow(b, 2) * pow(k, 4));`
10. Two implicit data type conversions occur. First, because `mass` is a `float`, a copy of the `int` value stored in `units` is promoted to a `float` before the multiplication operation is done. The result of `mass * units` will be a `float`. The second data type conversion occurs when the `float` result is promoted to a `double` in order to be stored in `double` variable `weight`.
11. 8
12. Either of these will work:  
`unitsEach = static_cast<double>(qty) / salesReps;`  
`unitsEach = qty / static_cast<double>(salesReps);`

13. `const int RATE = 12;`
14. `x += 5;`  
`total += subtotal;`  
`dist /= rep;`  
`ppl *= period;`  
`inv -= shrinkage;`  
`num %= 2;`
15. `east = west = north = south = 1;`
16. `int sum = 0;`
17. No, a named constant must be initialized at the time it is defined. It cannot be assigned a value at a later time.
18. `cout << fixed << showpoint << setprecision(2);`  
`cout << setw(8) << divSales;`
19. `cout << fixed << showpoint << setprecision(4);`  
`cout << setw(12) << profit;`
20. A) `cmath` B) `fstream` C) `iomanip`

Note: Once students understand that inputs from the keyboard should *always* be preceded by prompts, the `//` with `prompt` comment can be omitted from the pseudocode. Therefore, beginning with Chapter 3, we no longer include it.

21. *Input score1*  
*Input score2*  
*Input score3*  
`average = (score1 + score2 + score3) / 3.0`  
*Display average*
22. `discountPct = .15`  
*Input salesAmt*  
`amtSaved = salesAmt * discountPct`  
`amtDue = salesAmt - amtSaved`  
*Display amtSaved, amtDue*
23. *Input maxCredit*  
*Input creditUsed*  
`availableCredit = maxCredit - creditUsed`  
*Display availableCredit*
24. `PI = 3.14`  
`PRICE_PIZZA12 = 12.00`  
`PRICE_PIZZA14 = 14.00`  
`areaPizza12 = PI * (12 / 2)2`  
`areaPizza14 = PI * (14 / 2)2`  
`pricePerSqIn12 = PRICE_PIZZA12 / areaPizza12`  
`pricePerSqIn14 = PRICE_PIZZA14 / areaPizza14`  
*Display pricePerSqIn12, pricePerSqIn14*