Errata

Programming with C++, Second Edition, by John R. Hubbard
McGraw-Hill, 2000 (ISBN 0-07-135346-1)

The following corrections are for the first printing of the book.

You can tell which printing you have by looking at the sequence of consecutive integers on the back of the cover page (page ii). For example, if you see the sequence

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 PRS PRS 0 9 8 7 6 5 4 3 2 1 0

then you have the 1st printing. If you see the sequence

4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 PRS PRS 0 9 8 7 6 5 4 3 2 1 0

then you have the 4th printing.

Page iii:

Replace the last three lines of the penultimate paragraph with:

Geiger, Sergei Gorlatch, Chris Hanes, John B. Hubbard, Raana Jeelani, Dick Palas, Blake Puhak, Arni Sigurjonsson, Andrew Somers, Joe Troncale, Maureen Walker, Stefan Wentzig, and Nat Withers. Their editorial advice and debugging skills are gratefully appreciated.

Page 18:

Replace lines 8-9 with:

```
int answer;
cin >> answer;
```

Replace the next line of text with

to work as expected. If the value of the variable `answer` is 1,

Page 19:

Replace `<limits>` with `<climits>` on lines 2 and 5 of the code for Example 2.3.

Page 20:

Replace `<limits>` with `<climits>` on the first line after example output.

Page 28:

Replace the first "is" with "if" on the second line after Example 2.14.

Page 29:

Replace the last paragraph of Example 2.15 with

But this run attempts to solve the equation $x^2 + 10000000000x + 1 = 0$ and fails:

```
Enter the coefficients of a quadratic equation:
        a: 1
        b: 1e10
        c: 1
The equation is: 1*x*x + 1e10*x + 1 = 0
The solutions are:
        x1 = 0
        x2 = -1e10
Check:
        a*x1*x1 + b*x1 + c = 1
        a*x2*x2 + b*x2 + c = 1
```

The first solution, $x_1 = 0$, is obviously incorrect: the resulting quadratic expression $ax_1^2 + bx_1 + c$ evaluates to 1 instread of 0. The second solution, $x_2 = –1e10 = –10,000,000,000$ is even worse. The correct solutions are $x_1 = –0.00000\ 00000\ 99999\ 99999\ 99999\ 99519$ and $x_2 = 9,999,999,999.99999\ 99999$.

Page 45:

Replace "`Enter two`" with "`Enter three`" on the first line of the output block.

Page 48:

Insert after line 18:

Note that scores in the ranges 101 to 109 and -9 to -1 produce incorrect results. (See Problem 3.14.)

Page 51:

Add to the end of Problem 3.12: "(See Example 2.15 on page 28.)"

Add after Problem 3.13:

3.14 Correct Example 3.17 on page 47 so that it produces the correct response for all inputs.

Page 55:

Shift the first output block (lines 1-6) down to the end of the Solution to Problem 3.5.

Page 56:

Shift the first output block (lines 1-6) down to the end of the Solution to Problem 3.8.

Page 57:

Replace the solution to Problem 3.10 with

```
enum Winner {PLAYER1, PLAYER2, TIE};
int main()
{ int choice1, choice2;
  Winner winner;
  cout << "Choose rock (0), paper (1), or scissors (2):" << endl;
  cout << "Player #1: ";
  cin >> choice1;
  cout << "Player #2: ";
  cin >> choice2;
  switch (choice2 - choice1)
  { case  0:
      winner = TIE;
      break;
    case -1:
    case  2:
      winner = PLAYER1;
      break;
    case -2:
    case  1:
      winner = PLAYER2;
  }
  if (winner == TIE) cout << "\tYou tied.\n";
  else if (winner == PLAYER1) cout << "\tPlayer #1 wins." << endl;
  else cout << "\tPlayer #2 wins." << endl;
}
```

Insert after the last line:

3.14 A corrected version of Example 3.17 on page 47:

```
int main()
{ // reports the user's grade for a given test score:
  int score;
  cout << "Enter your test score: ";
  cin >> score;
  if (score > 100 || score < 0)
    cout << "Error: that score is out of range.\n";
  else
    switch (score/10)
    { case 10:
      case  9: cout << "Your grade is an A.\n";  break;
      case  8: cout << "Your grade is a B.\n";   break;
```

```
      case  7: cout << "Your grade is a C.\n";   break;
      case  6: cout << "Your grade is a D.\n";   break;
      default: cout << "Your grade is an F.\n";  break;
    }
  cout << "Goodbye." << endl;
}
```

```
Enter your test score: 103
Error: that score is out of range.
Goodbye.
```

```
Enter your test score: 93
Your grade is an A.
Goodbye.
```

```
Enter your test score: -3
Error: that score is out of range.
Goodbye.
```

Page 58:

Replace the solution to Problem 3.11 with

```
enum Winner {PLAYER1, PLAYER2, TIE};
int main()
{ int choice1, choice2;
  cout << "Choose rock (0), paper (1), or scissors (2):" << endl;
  cout << "Player #1: ";
  cin >> choice1;
  cout << "Player #2: ";
  cin >> choice2;
  int n = (choice1 - choice2 + 3) % 3;
  Winner winner = ( n==0 ? TIE : (n==1?PLAYER1:PLAYER2) );
  if (winner == TIE) cout << "\tYou tied.\n";
  else if (winner == PLAYER1) cout << "\tPlayer #1 wins." << endl;
  else cout << "\tPlayer #2 wins." << endl;
}
```

Change "`main()`" to "`int main()`" on the third line of code in Problem 3.12.

Page 65:

Indent lines 6, 8, and 10.

Replace lines 8-9 of the code for Example 4.9 with

```
{ f *= ++i;
  cout << ", " << f;
```

Page 67:

Replace lines 4-5 with:

```
{ f *= i;
  cout << ", " << f;
```

Page 68:

Replace penultimate line of code in Example 4.16 with

```
cout << "min = " << min << endl;
```

Page 76:

Add the following after the second line of code in Example 4.26 with

```
using namespace std;
```

Page 77:

Add the following after the second line of code in Example 4.27 with

```
using namespace std;
```

Page 78:

Add the following after the fourth line of code in Example 4.28 with

```
using namespace std;
```

Page 79:

Add the following after the fourth line of code in Example 4.29 with

```
using namespace std;
```

Page 84:

Replace the second line of code in the Solution to Problem 4.10 with

```
for (q = 0, r = n; r >= d; q++)
```

Page 85:

Add the following after the second line of code in the Solution to Problem 4.12:

```
using namespace std;
```

Replace lines 5-6 of the code in the Solution to Problem 4.13 with

```
int n=1;
while (n*n<x)
  ++n;
```

Page 94:

Replace the first line of code after Example 5.8 with

```
$ c++ -c max.c
```

Page 114:

Replace "it" with "its" on the first line of the Answer to Review Question 5.2.

Page 140:

Replace "`main`" with "`int main`" on the first line.

Page 146:

Replace the last sentence in Problem 6.34 with

Extend the program from Problem 6.33 so that it prints the "curved" grade for each of the test scores read.

Page 149:

Replace the third line of code in the Solution to Problem 6.9 with

```
int j=0;
```

Page 155:

Replace this page with

6.33

```
int main()
{ double x[] = { 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8, 9.9 };
  int n=8;
  print(x,n);
  double m = mean(x,n);
  double s = stdev(x,n);
  cout << "mean = " << m << ", std dev = " << s << endl;
  for (int i=0; i<n; i++)
    cout << "x[" << i << "] = " << x[i]
         << ", z[" << i << "] = " << (x[i] - m)/s << endl;
}
```

6.34

```
int main()
{ double x[] = { 2.5, 4.5, 6.3, 6.7, 7.2, 7.5, 7.8, 9.9 };
  int n=8;
  print(x,n);
  double m = mean(x,n);
  double s = stdev(x,n);
  cout << "mean = " << m << ", std dev = " << s << endl;
```

```
  for (int i=0; i<n; i++)
  { double z = (x[i] - m)/s;
    cout << "x[" << i << "] = " << x[i]
         << ", z[" << i << "] = " << z;
    if (z >= 1.5) cout << " = A" << endl;
    else if (z >= 0.5) cout << " = B" << endl;
    else if (z >= -0.5) cout << " = C" << endl;
    else if (z >= -1.5) cout << " = D" << endl;
    else  cout << " = F" << endl;
  }
}
```

6.35

```
void build_pascal(int p[][SIZE], int n)
{ assert(n > 0 && n < SIZE);
  for (int i=0; i<SIZE; i++)
    for (int j=0; j<SIZE; j++)
      if (i>n || j>i) p[i][j] = 0;
      else if (j==0 || j==i) p[i][j] = 1;
      else p[i][j] = p[i-1][j-1] + p[i-1][j];
}
```

6.36

```
double max_of_col(Matrix m, int n, int j)
{ double max=m[0][j];
  for (int i=1; i<n; i++)
    if (m[i][j]>max) max = m[i][j];
  return max;
}
double minimax(Matrix m, int n)
{ assert(n>0 && n < SIZE);
  double minimax=max_of_col(m,n,0);
  for (int j=1; j<n; j++)
  { double mm = max_of_col(m,n,j);
    if (mm<minimax) minimax = mm;
  }
  return minimax;
}
```

Page 165:

Delete the last line of output in Example 7.11.

Replace lines 5-6 in the code for Example 7.12 with

```
{ int j;
  for (j = 0; j < n2; j++)
    if (p1[j] != a2[j]) break;
```

Page 176:

Replace "*Random*" with "*Perfect*" in Problem 7.16.

Replace `sin()` with `cos()` in Problem 7.21*b*.

On the fifth line of Problem 7.23, replace "the call `root(square,1,2,100)`" with

if $f(x) = x^2 - 2$, then `root(f,1,2,100)`

Page 177:

Replace "to which `n` points" with "to which `p` points" in the Answer to Review Question 7.5.

Page 180:

Replace the third line of code in the Solution to Problem 7.3 with

```
for (int i = 0; i < n; i++)
```

Replace the third line of code in the Solution to Problem 7.4 with

```
int main()
```

Page 181:

Replace the first line of code in the Solution to Problem 7.5 with

```
#include <iostream>
```

Replace the fifth line of code in the Solution to Problem 7.5 with

```
int main()
```

Page 182:

Replace the ninth line of code in the Solution to Problem 7.6 with

```
int main()
```

Append to the bottom of the page

Solutions to Problems 7-7-7.24 are available on-line at `ProjectEuclid.net`.

Page 186:

Replace "**Strings**" with "**C-Strings**" in the title of Example 8.3.

Replace "Ctrl+Z (Ctrl+D on a UNIX or Macintosh computer)" with "Ctrl+Z+Enter+Enter" in the sixth line from the bottom of the page.

Page 187:

Change the title of Example 8.5 to

The `cin.getline()` Function with Three Parameters

Replace "`20`" with "`80`" on the second and fourth lines of code in Example 8.5.

Page 188:

Replace the output block with

```
Once upon a midnight dreary, while I pondered, weak and weary,
        [Once upon a midnight dreary]
        [ while I pondered]
        [ weak and weary]
Over a many quaint and curious volume of forgotten lore,
        [
Over a many quaint and curious volume of forgotten lore]
^Z
        [
]
```

Page 192:

Replace "`!`" with "`!=`" on line 7 of the code in Example 8.11.

Page 197:

Replace the first two lines of code in Example 8.18 with

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

Replace the first two lines of code in Example 8.19 with

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

Page 199:

Replace the first two lines of code in Example 8.20 with

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

Page 202:

Replace the fifth line of code in Problem 8.1 with

```
name[count++] = buffer;
```

Page 204:

Replace the last line of code in the Answer to Review Question 8.2.*b* with

```
char* s = "Hello";  // defines s as a pointer to a string constant
```

Page 205:

Replace the second line of code in the Solution to Problem 8.2 with

```
{ char* p;  for (p=s1; *s2; )
```

Replace the second line of code in the Solution to Problem 8.3 with

```
{ char* end;  for (end=s1; *end; end++)
```

Replace the fourth line of code in the Solution to Problem 8.3 with

```
{ char* p;  for (p=s2; *p && p-s2<n; )
```

Replace the penultimate line with

```
bool isvowel(char c)
```

Page 206:

Shift the first line back to the bottom of page 205.

Replace the first line of code with

```
bool pluralize(char*);
```

Replace “25 characters” with “20 characters” on the first line of the Solution to Problem 8.5.

Replace the last two lines with

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

Page 208:

Replace the lines 2-3 of code in the Solution to Problem 8.6 with

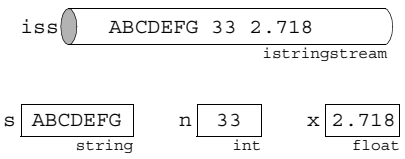
```
{ char* end, temp;
  for (end = s; *end; end++)
    ;    // find end of s
```

Pages 219-220:

Replace the lines from Example 9.8 to the Review Questions with

EXAMPLE 11.8 Using an Output String Stream

This program creates four objects: a character string `s`, an integer `n`, a floating-point number `x`, and an output string stream `oss`:




```
#include <iostream>
#include <sstream>
#include <string>
using namespace std;
void print(ostringstream&);
int main()
{ string s="ABCDEFG";
  int n=33;
  float x=2.718;
  ostringstream oss;
  print(oss);
  oss << s;
  print(oss);
  oss << " " << n;
  print(oss);
  oss << " " << x;
  print(oss);
```

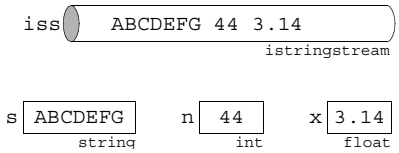
```
}
void print(ostringstream& oss)
{ cout << "oss.str() = \"" << oss.str() << "\"" << endl;
}
```

```
oss.str() = ""
oss.str() = "ABCDEFG"
oss.str() = "ABCDEFG 33"
oss.str() = "ABCDEFG 33 2.718"
```

The output string stream object `oss` acts like the output stream object `cout`: the values of the string `s`, the integer `n`, and the number `x` are written to it by means of the insertion operator `<<`.

While the internal object `oss` is like an external text file, its contents can be accessed as a `string` object by the call `iss.str()`.

EXAMPLE 11.9 Using an Input String Stream



This program is similar to the one in Example 11.8 except that it reads from an input string stream `iss` instead of writing to an output string stream.:

```
void print(string&,int,float,istringstream&);
int main()
{ string s;
  int n=0;
  float x=0.0;
  istringstream iss("ABCDEFG 44 3.14");
  print(s,n,x,iss);
  iss >> s;
  print(s,n,x,iss);
  iss >> n;
  print(s,n,x,iss);
  iss >> x;
  print(s,n,x,iss);
}
void print(string& s, int n, float x, istringstream& iss)
{ cout << "s = \"" << s << "\", n = " << n << ", x = " << x
       << ", iss.str() = \"" << iss.str() << "\"" << endl;
}
```

```
s = "", n = 0, x = 0, iss.str() = "ABCDEFG 44 3.14"
s = "ABCDEFG", n = 0, x = 0, iss.str() = "ABCDEFG 44 3.14"
s = "ABCDEFG", n = 44, x = 0, iss.str() = "ABCDEFG 44 3.14"
s = "ABCDEFG", n = 44, x = 3.14, iss.str() = "ABCDEFG 44 3.14"
```

The input string stream object `iss` acts like the input stream object `cin`: values for the string `s`, the integer `n`, and the number `x` are read from it by means of the extraction operator `>>`. But the `iss` object also acts like an external file: reading from it does not change its contents.

Page 224:

Replace "`HindArabic`" with "`HinduArabic`".

Page 228:

Replace line 2 of the code for the Solution to Problem 9.8 with

```
{ ifstream in("Pr0907.in");
  string s;
```

Replace lines 7-13 of the code for the Solution to Problem 9.8 with

```
  while (in >> s)
  { reduce(s);
    if (s.length() == 0) continue;
```

```
        ++words;
        in.get(c);
        if (c == '\n') ++lines;                          // count line
        for (i=0; i<n; i++)
```

Page 229:

Replace line 12 of the code for the Solution to Problem 9.10 with

```
else s += string(d2, 'C');
```

Page 230:

Replace the last 8 lines of the code for the Solution to Problem 9.11 with

```
int HindArabic(string s)
{ int n0=0, n1=0, n=0;
  for (int i=0; i<s.length(); i++)
  { n0 = n1;
    n += n1 = v(s,i);
    if (n1>n0) n -= 2*n0;
  }
  return n;
}
```

Page 231:

Replace the Solution to Problem 9.16 with

```
bool more(ifstream& fin, string& s)
{ if (getline(fin, s)) return true;
  else return false;
}
bool copy(ofstream& fout, ifstream& fin, string& s)
{ fout << s << endl;
  cout << s << endl;
  return more(fin,s);
}
int main()
{ ifstream fin1("Democrats.dat");
  ifstream fin2("Republicans.dat");
  ofstream fout("Presidents.dat");
  string s1, s2;
  bool more1 = more(fin1, s1);
  bool more2 = more(fin2, s2);
  while (more1 && more2)
    if (s1 < s2) more1 = copy(fout, fin1, s1);
    else more2 = copy(fout, fin2, s2);
  while (more1)
    more1 = copy(fout, fin1, s1);
  while (more2)
    more2 = copy(fout, fin2, s2);
  fout << endl;
}
```

Page 234:

Append a right brace "}" to the end of line 6 of the code for Example 10.2.

Page 235:

Append a right brace "}" to the end of line 6 of the code for Example 10.3.

Page 236:

Replace "`Ratio`" with "`class Ratio`" on the first line of code in Example 10.4.

Append a right brace "}" to the end of line 6 of the code in Example 10.4.

Page 240:

Replace "`main`" with "`int main`" in the code for Example 10.9.

Page 241:

Replace "`Ratio`" with "`class Ratio`" on the first line of code in Example 10.10.

Replace "`main`" with "`int main`" in the code for Example 10.10.

Page 242:

Replace "`main`" with "`int main`" in the code for Example 10.11.

Page 244:

Replace "`main`" with "`int main`" in the code for Example 10.12.

Page 245:

Replace line 6 with

```
for ( ; p; p = p->next )
```

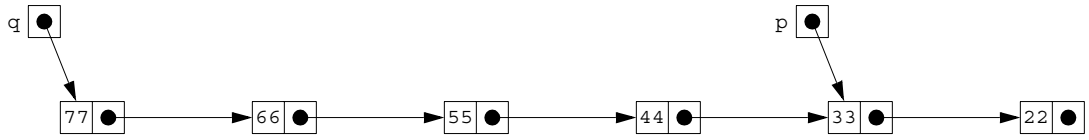
Replace the second line of the output block with

```
77 -> 66 -> 55 -> 44 -> 33 -> 22 -> *
```

Replace the last two lines of the third paragraph after the output block with

last node constructed) and continuing until `p` is `NUL`.

Replace the figure with



Page 246:

Replace "`main`" with "`int main`" in the code for Example 10.14.

Replace "`main`" with "`int main`" in the code for Example 10.15.

Page 248:

Replace "`main`" with "`int main`" in the code for Example 10.16.

Replace "the `num()` function" with "the `numWidgets()` function" on the first line of text.

Page 251:

Replace the first two lines of code in the Solution to Problem 10.1 with

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

Replace "`int isEmpty()`" with "`bool isEmpty()`" in the Solution to Problem 10.2.

Replace "`int isFull()`" with "`bool isFull()`" in the Solution to Problem 10.2.

Page 252:

Replace line 4 of the code for the Solution to Problem 10.4 with

```
#include <ctime>
using namespace std;
```

Replace "`seed(long seed`" with "`void seed(long seed`" in the Solution to Problem 10.4.

Page 253:

Replace each of the three occurrences of "12" with "10" on lines 3-5.

Replace line 3 of the code for the Solution to Problem 10.5 with

```
Person(char* =0, int =0, int =0);
```

Replace line 14 of the code for the Solution to Problem 10.5 with

```
Person::Person(char* name, int yob, int yod)
```

Page 257:

Replace the last line of code before Example 11.2 with

```
Ratio& operator=(const Ratio& r)
```

Page 260:

Replace "`int`" with "`bool`" in the second line of code in Example 11.7.

Page 261:

Replace "`int`" with "`bool`" on line 5.

Page 264:

Replace the last output line in Example 11.10 with

```
P = 22/7, PI = 3.14268
```

Page 266:

Remove the first two lines of code in Example 11.13.

Replace the last six lines of code in Example 11.13 with

```
ostream& operator<<(ostream& ostr, const Ratio& r)
{ return ostr << r.num << "/" << r.den;
}
int& Ratio::operator[](int i)
{ if (i == 1) return num;
  else return den;
}
```

Page 267:

Delete Problem 11.5.

Replace "Example 10.1 on page 233" with "Problem 10.1 on page 249" in Problems 11.7-11.10.

Page 268:

Swap the Answers to Review Questions 11.2 and 11.3.

Replace "`int`" with "`bool`" in the fourth line of code for the Solution to Problem 11.1.

Replace the penultimate line with

```
Ratio operator-(const Ratio& x, const Ratio& y)
```

Page 269:

Replace the fourth line of code with

```
Ratio operator-(const Ratio& x)
```

Replace "`int`" with "`bool`" in line 22 and line 36.

Replace the eleventh line of code with

```
Ratio::Ratio(int n, int d) : num(n), den(d)
```

Page 270:

Replace "`int`" with "`bool`" in line 1.

Replace the fifth line of code from the bottom with

```
Vector::Vector(int sz, double t) : size(sz)
```

Page 271:

Replace "`_num`" with "`num`" and "`_den`" with "`den`" in the Solution to Problem 11.3.

Page 272:

Replace "`_num`" with "`num`" and "`_den`" with "`den`" in the Solution to Problem 11.4.

Page 280:

Replace "`double a`" with "`string a`" in the third line of code.

Page 281:

Replace "`z` is instantiated" with "that of `x`" in the fourth line of text after the code in Example 12.8.

Page 284:

Replace the code for Example 12.11 with

```
class Person
{ public:
    Person(char* s) { name = new char[strlen(s)+1]; strcpy(name, s);}
    void print() { cout << "My name is " << name << ".\n"; }
  protected:
    char* name;
};

class Student : public Person
{ public:
    Student(char* s, float g) : Person(s), gpa(g) { }
    void print()
    { cout << "My name is " << name << " and my G.P.A. is "
           << gpa << ".\n"; }
  private:
    float gpa;
};

class Professor : public Person
{ public:
    Professor(char* s, int n) : Person(s), publs(n) { }
    void print()
    { cout << "My name is " << name
           << " and I have " << publs << " publications.\n"; }
  private:
    int publs;
};

int main()
{ Person* p;
  Person x("Bob");
  p = &x;
  p->print();
  Student y("Tom", 3.47);
  p = &y;
  p->print();
  Professor z("Ann", 7);
  p = &z;
  p->print();
}
```

Page 286:

Replace “3992” with “4092” on the fourth text line.

Page 288:

Remove the semicolons at the ends of code lines 13 and 22.

Page 293:

Replace “`KING`” with “`KING, ACE`” in the second line of code in the Solution to Problem 12.1.

Page 296:

Delete “`=5`” on line 5.

Replace “`i/13`” with “`i%4`” on line 17.

Insert the following line above the first line of code in the Solution to Problem 12.2:

```
const double PI=3.14159265358979;
```

Page 297:

Replace "`pi`" with "`PI`" on lines 6, 7, 16, and 26.

Page 301:

Replace lines 3-4 of the code for Example 13.2 with (thus deleting "`T temp;`")

```
{ for (int i=1; i<n; i++)
```

Page 305:

Replace "`T[size]`" with "`T[n]`" in the fourth line of code.

Replace "`T[size]`" with "`T[v.size]`" in the fifth line of code.

Page 309:

Replace "Example 10.3 on page 235" with "Example 10.13 on page 244" in the first line of Section 13.7.

Page 310:

Replace lines 4-5 with

Note that the class `List<T>` is declared here to be a `friend` of the `ListNode` class. This will allow the member functions of the `List` class to access the protected members of the `Node` class. For this statement to compile, some compilers require the following *forward reference* to precede the `ListNode` template definition:

```
template<class T>
class List;
```

This simply tells the compiler that the identifier `List` will be defined later as a class template.

Replace "`int`" with "`bool`" on line 8 of the code for Example 13.7.

Page 313:

Replace "`virtual int`" with "`virtual void`" in the fourth line of code.

Page 318:

Indent line 11.

Page 323:

Indent lines 12-16.

Page 354:

Replace line 24 with

```
reverse_iterator rbegin();
```

Replace line 26 with

```
reverse_iterator rend();
```

Page 355:

Replace "`reverse_iterator It`" with "`iterator It`" on the fourth line of Example D.1.

Page 372:

Delete the (duplicate) sixth line from the bottom of the page.

Page 380:

Adjust the shading for the output for Example E.30.

Page 385:

Replace "`29`" woth "`30`" on the last line of the output block for Example E.43.

Page 386:

Replace "`29`" woth "`30`" on the last line of the output block for Example E.44.

Page 402:

Replace the title line for Example G.2 with

EXAMPLE K.10 Using Input Manipulators

Replace "`cin >> n`" with "`cin >> oct >> n`" on line 4 of Example G.2.

Replace "`cin >> n`" with "`cin >> dec >> n`" on line 9 of Example G.2.
Replace "`cin >> n`" with "`cin >> hex >> n`" on line 14 of Example G.2.

Page 412:

Insert the Index entries:
`cstdlib`, 76
`ctime`, 78

Page 414:

Insert the Index entries:
`srand()`, 77
`time()`, 78

Page 418:

Replace "401" with "400" in the index listing for "`peek()`".

Page 419:

Insert the Index entry:
`rand()` function, 76
Delete the Index entry for "`setw`".

Page 420:

Insert the Index entry:
`srand()` function, 77
Delete the Index entry for "`strcat()`".
Add page numbers 185, 196, 199, 201 to the Index entry for "`strcat()` function".
Delete the Index entry for "`strchr()`".
Add page numbers 199, 201 to the Index entry for "`strchr()` function".
Delete the Index entry for "`strcpy()`".
Add page numbers 185, 196, 199, 201, 202, 205 to the Index entry for "`strcpy()` function".

Page 421:

Insert the Index entries:
`time()` function, 78
Type definitions, 138
Delete the Index entry for "`typedef`".
Add page numbers 138, 326 to the Index entry for "`typedef` keyword".

Also include the corrections listed below for later printings.

The following corrections are for the second printing of the book.

Page 23:

In the output for EXAMPLE 2.7:

change 55 to 54
change 75 to 74
change 35 to 34
change 1100 to 1080

Page 32:

Replace review questions 2.1-2.4 with:

2.1 What is the difference between the `short`, `int`, and `long` types?

2.2 What is the difference between the following two definitions?

a. `enum Direction { NORTH, EAST, SOUTH, WEST };`

b. `const int NORTH=0, EAST=1, SOUTH=2, WEST=3;`

2.3 What is the main use of the `unsigned` integer types?

2.4 What is the difference between the pre-increment `++` operator and the post-increment `++` operator?

Page 33:

Replace the answers to review questions 2.1-2.4 with:

2.1 All three are integer types. But `short` uses 16 bits, `int` uses 32 bits, and `long` uses 64 bit.

2.2 The only difference is that the four constants that are defined by the `enum` statement have type `Direction` instead type `int`.

2.3 The `unsigned` integer types are used primarily for bit strings.

2.4 If used stand-alone, like this

```
++n;
i++;
```

then there is no difference; they both simply add 1 to the value of the variable. But if used within a larger expression, like this

```
m = ++n;
cout << i++;
```

then the difference is that the pre-increment will add 1 first and then use the resulting value within the larger expression, while the post-increment will add 1 after using the current value within the larger expression.

Also include the corrections listed below for later printings.

The following corrections are for the third printing of the book.

Pages 9-5:

See the **Pages9-15.pdf** file online.

Page 22:

In the inline comments in EXAMPLE 2.6:

replace "`multiplies n by 3`" with "`multiplies n by 2`"

replace "`divides n by 9`" with "`divides n by 3`"

Pages 32-35:

See the **Pages32-37.pdf** file online.

Page 132:

In EXAMPLE 6.10, move the const definition out of the `main()` function, changing the first five lines of code to:

```
void read(int[],int&);
void print(int[],int);
const int MAXSIZE=100;
int main()
{ int a[MAXSIZE]={0}, size;
```

Page 173:

In Review Question 7.15, replace "`0x3fffd00`" with "`0x3fffd00`, and `n` is stored directly ahead of `m` in byte `0x3fffcfc`".

The following corrections are for the fourth printing of the book.

Pages 22-26:
Replace those five pages with:

Pages 41-45:
Replace those five pages with: