

---

**Page 17:** Paragraph 6, in the command lines, the comment at the end of the first line, *Compile prog1.m & call it prog1* should not be in bold face (italics, but not bold).

**Page 26** - Exercise 6 - 2nd line of program:

```
int main (int argc, const char * argv[])
```

should be:

```
int main (int argc, const char * argv[])
```

**Page 29:** The instance `yourCar` is being used with some examples of message expressions. The last example, setting `currentMileage`, is using the wrong instance:

```
currentMileage = [suesCar currentOdometer];
```

should be:

```
currentMileage = [yourCar currentOdometer];
```

**Page 38:** An `@` is missing in the `NSLog` statement:

```
NSLog ("%i/%i", numerator, denominator);
```

should be:

```
NSLog (@"%i/%i", numerator, denominator);
```

**Page 50:** In the 4th paragraph the word "hexidecimal" should be "hexadecimal"

**Page 70:** In the third code snippet, in the middle of the page change the hexadecimal at the end of the second line:

```
0x5ab0
```

should be:

```
0x5ad0
```

**Page 71:** In the third paragraph, delete the sentence that starts "The sequence of statements that follows..."

**Page 97:** In Exercise 4, change the program to be modified from Program 5.2. to Program 5.3.

**Page 109:** The expression that should have no use of parenthesis (except the outermost) contains an extraneous closing parenthesis in the middle:

```
if(year%4==0&&year%100!=0) | | year%400==)
```

should be:

```
if(year%4==0&&year%100!=0 | | year%400==)
```

**Page 115:** At the end of the first paragraph after the colon, delete: `[[STYLE FIRST]]`

**Page 146:** At the end of the first paragraph, change "argument" to "receiver":

"Later in this chapter, you'll redefine the `add:` method so that `add:` does not affect the value of its argument.

should be:

"Later in this chapter, you'll redefine the `add:` method so that `add:` does not affect the value of its receiver.

**Page 153:** The code snippet with the method `setTo:over:` is missing the square bracket at the end:

```
[result setTo: numerator * f.denominator + denominator * f.numerator  
        over: denominator * f.denominator;
```

*should be:*

```
[result setTo: numerator * f.denominator + denominator * f.numerator  
        over: denominator * f.denominator];
```

**Page 155:** In code listing at bottom of page, after the line

```
-(void) print;
```

insert a new line:

```
-(void) setTo: (int) n over: (int) d;
```

*Listing should read:*

```
@property int numerator, denominator;  
  
-(void) print;  
-(void) setTo: (int) n over: (int) d;  
-(double) convertToNum;  
-(Fraction *) add: (Fraction *) f;  
-(void) reduce;  
@end
```

**Page 163:** In Program 8.2 delete line that reads

```
#import <stdio.h>
```

**Page 170:**

```
-(Point *) origin
```

*should be:*

```
-(XYPoint *) origin
```

**Page 171:** Program 8.4 - left-align the line that reads `[myRect release];` so it lines up with the line below it.

**Page 176:** After the heading Program 8.6 Output, the line "Clearly, the message" should be formatted as body text not as code.

**Page 185:** Exercise 7 - add the following note to the end of the exercise:

Note: You should use `printf` to draw your characters, since `NSLog` will display a new line each time it's called.

**Page 188:** In paragraph 1 change

Exercise 6 from Chapter 4, "Data Types and Expressions."

to

Exercises 6 and 7 from Chapter 7, "More on Classes."

**Page 191:** The reference in the Note should be to Chapter 17, "Memory Management" rather than to Chapter 18.

---

**Page 196:** Paragraph 2

Change `Object` to `NSObject`

**Page 197:** First sentence, change it to read:

“To see if the variable `myFract` is a `Fraction` class object, you test...”

**Page 216:** Program 10.3

Insert the following line before the first line of the listing:

```
#import <Foundation/Foundation.h>
```

**Page 222:** In Exercise 1 remove the extra colon after `w` in the line of code:

```
-(Rectangle *) initWithWidth: (int) w: andHeight: (int) h;
```

*should be:*

```
-(Rectangle *) initWithWidth: (int) w andHeight: (int) h;
```

**Page 226:** Delete the following line at the top of the page:

```
#import <stdio.h>
```

**Page 251:** In the sixth paragraph (which begins “As already mentioned...”), second sentence, replace reference to `printf` to `NSLog`:

“You might have many `printf` calls...”

*should be*

“You might have many `NSLog` calls...”

**Page 255:** In the Note change the sentence to read:

“There are ways to work with multibyte characters at the C level, but Foundation provides a much more elegant solution with its `NSString` class.”

**Page 255:** 2nd paragraph, 2nd sentence. Change the sentence to read:

“If you're working with large arrays of data, for example, you might want to use the built-in data structures of C instead of the array objects of Foundation...”

**Page 263:** Program 13.3 Output should say "Program 13.3 Output," not "Program 13.4 Output"

**Page 273:** In the second paragraph, second sentence:

"... so that 04 displays for the year."

*should be:*

"... so that 09 displays for the year."

**Page 310:** In paragraph 3 `strotol` should be `strtol`

**Page 325:** In Table 15.1 the retrieval instance method for the `Int` should read `intValue` instead of `intValueunsigned`

**Page 326:** In the third paragraph, the middle of the second sentence, add space between “of” and “Table” in “first column ofTable 15.1”.

**Page 327,** After heading Program 15.2 Output, the line “In the line” should be formatted as normal text not code.

**Page 356:** First paragraph of text, third line, remove “the” after “list” and change “that” after “verify” to “the”:  
“... and the list method is used to list the and verify that contents of the...”

*should be:*

“... and the list method is used to list and verify the contents of the...”

**Page 359:** In Program 15.12, the return type of the constructor should be id. It should read:

```
-(id) initWithName: (NSString *) name;
```

**Page 361:** In Program 15.12, the declaration of `myCard` is missing a semi-colon. It should read:

```
AddressCard *myCard;
```

**Page 366:** Table 15.4, last row of code, change `automically` to `atomically`:

```
-(BOOL) writeToFile: path automically: (BOOL) flag
```

*should be*

```
-(BOOL) writeToFile: path atomically: (BOOL) flag
```

**Page 376:** In Exercise 10 change the first sentence to read:

“Write a program that takes an `NSArray` of `NSNumber` objects (where each `NSNumber` represents an integer) and produces a frequency chart that lists each integer and how many times it occurs in the array.”

**Page 383:** End of first paragraph, 8 exabytes should be 8 *billion* gigabytes, not 8 thousand gigabytes.

**Page 419:** Fourth paragraph, line 2, remove the “of” after “understand”:

“..., you will thoroughly understand of the autorelease pool and how it works.”

*should be:*

“..., you will thoroughly understand the autorelease pool and how it works.”

**Page 436:** Near the end of the code listing at the top of the page, replace the two lines that read

```
if ([glossary writeToFile: @"glossary" atomically: YES  
    encoding: NSUTF8Encoding error: nil] == NO)
```

*with this one line:*

```
if ([glossary writeToFile: @"glossary" atomically: YES] == NO)
```

**Page 443:** In line 1 add @ before `AddressBookBookName`:

```
[encoder ..... "AddressBookBookName"
```

*should be*

```
[encoder ..... @"AddressBookBookName"
```

**Page 443:** Move the following line in the listing:

```
NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
```

before this line:

```
NSString *aName = @"Julia Kochan";
```

**Page 447:** First paragraph, line 7:

decodeInt:forKey: should be decodeIntForKey:

**Page 448:** replace line of code with two new lines:

```
if ([dataArea writeToFile: @"myArchive" atomically: YES  
    encoding::NSUTF8Encoding error: nil] == NO)
```

*should be:*

```
if ([dataArea writeToFile: @"myArchive" atomically: YES] == NO)
```

**Page 449:** In the middle of the second paragraph, the line of code:

```
[dataArea writeToFile: @"myArchive" atomically: YES  
    encoding::NSUTF8Encoding error: nil]
```

*should be:*

```
[data writeToFile: @"myArchive" atomically: YES]
```

**Page 449:** In the second paragraph's last sentence:

```
writeToFile:atomically:encoding:error:
```

*should be*

```
writeToFile:atomically:
```

**Page 449:** In the third paragraph:

```
writeToFile:atomically:encoding:error:
```

*should be*

```
writeToFile:atomically:
```

**Page 450:** Code snippet:

```
Return (1);
```

*should be:*

```
return 1;
```

**Page 451:** Five lines and nine lines from the bottom:

```
NSLog ( ".....")
```

*should be:*

```
NSLog ( @".....")
```

**Page 451:** In program 19.11 at the very bottom:

```
[pool drsin];
```

*should be:*

```
[pool drain];
```

**Page 457:** First paragraph, third line, "iTouch" should be "iPod Touch"

**Page 459:** First paragraph, last sentence, change "you'll learned" to "you learned"

**Page 468:** In the third paragraph "as shown in Figure 21.8" should be "as shown in Figure 21.9"

**Page 481:**

```
-(void) processOp: (char) op;
```

*should be:*

```
-(void) processOp: (char) theOp;
```

**Page 484:** In the code snippet, `dealloc` method:

```
[myCalculator dealloc];
```

*should be:*

```
[myCalculator release];
```