

Errata for *Elements of Programming*  
Alexander Stepanov and Paul McJones  
Addison-Wesley Professional, 2009  
[www.elementsofprogramming.com](http://www.elementsofprogramming.com)

This document was last edited January 26, 2010.

## Second printing, October 2009

### Chapter 1

Page 12, line 21: Change  $x \mapsto \mathcal{F}(x)$  to  $x \mapsto \mathcal{F}(x)$ . (Reported by Abraham Sebastian.)

### Chapter 2

Page 25, `convergent_point`: Add this precondition:

*// Precondition:  $(\exists n \in \text{DistanceType}(F)) n \geq 0 \wedge f^n(x_0) = f^n(x_1)$*

Page 26: Change Exercise 2.3 to:

The precondition of `convergent_point` ensures termination. Implement an algorithm `convergent_point_guarded` for use when that precondition is not known to hold, but there is an element in common to the orbits of both `x0` and `x1`.

(Reported by Abraham Sebastian.)

### Chapter 3

Page 33, bottom of page: Change the case for `n` odd from  $(a^2)^{n/2} a$  to  $(a^2)^{\lfloor n/2 \rfloor} a$ . (Reported by Ryan Ernst.)

Page 44, line 5: Change `xx-k+1` to `xn-k+1`. (Reported by Foster Brereton.)

### Chapter 4

Page 49, the *Relation* concept should be:

$$\begin{aligned} \text{Relation}(\text{Op}) &\triangleq \\ &\text{HomogeneousPredicate}(\text{Op}) \\ &\wedge \text{Arity}(\text{Op}) = 2 \end{aligned}$$

(Reported by Foster Brereton.)

Page 54, line 12: Change “jth largest of k arguments” to “jth element from k arguments according to the given ordering.” (Reported by Abraham Sebastian.)

## Chapter 5

Page 66: Change the first sentence to: “A transformation is called an *inverse operation* of a binary operation with respect to a given element (usually the identity element of the binary operation) if it satisfies the following:”. (Reported by Mike Spertus.)

Page 74, line 13: Change the final colon to a period. (Reported by Ryan Ernst.)

## Chapter 6

Page 101, `find.n`: the precondition should be `readable_weak_range(f, n)` rather than `weak_range(f, n)`.

## Chapter 7

Page 121, `reachable`: the requirement should be *BidirectionalBifurcateCoordinate* rather than *BifurcateCoordinate*, and the precondition should be `tree(x)` rather than `tree(c)`. (The requirement error was reported by the GHC compiler when the concepts and functions in this chapter were translated to Haskell type classes and functions.)

## Appendix 2

Page 237: replace the production for `control_statement` with:

```
control_statement = return | conditional | switch | while | do
                  | compound | break | goto.
```

(Reported by Carla Villoria Burgazzi.)

---

## First printing, June 2009

### Chapter 1

Page 14, `regular_unary_function`: the *f* to the left of  $\mapsto$  should be *f*. (Reported by Mark Ruzon.)

## Chapter 2

Page 18, `power_unary` precondition: the superscript should be `i` rather than `n`:

```
// Precondition:  $n \geq 0 \wedge (\forall i \in \mathbb{N}) 0 < i \leq n \Rightarrow f^i(x)$  is defined
```

(Reported by Alexei Nekrassov.)

Page 24, fourth line: Change “`q ≥ 0`” to “`q > 0`” and “when `slow` enters the cycle” to “when it collides with `slow`”. (Reported by Bob English and John Banning.)

## Chapter 4

Page 52, the `weak_ordering` property should be:

```
property(R : Relation, E : Relation) requires(Domain(R) = Domain(E))  
weak_ordering : R  
  r ↦ transitive(r) ∧ (∃e ∈ E) equivalence(e) ∧  
      (∀a, b ∈ Domain(R)) exactly one of the following holds:  
      r(a, b), r(b, a), or e(a, b)
```

(Reported by Eugene Kirpichov.)

Page 59, `select_2_5_ab_cd`: the first statement should be:

```
compare_strict_or_reflexive<(ia < ic), R> cmp;
```

(Reported by Hao Song.)

## Chapter 5

Page 74: the sentence “While we believe that there is no logarithmic time, constant-space algorithm for remainder on Archimedean monoids, an iterative constant-space algorithm exists when we can divide by 2.”<sup>3</sup> should be changed to “Floyd and Knuth [1990] give a constant-space algorithm for remainder on Archimedean monoids that performs about 31% more operations than `remainder_nonnegative`, but when we can divide by 2 an algorithm exists that does not increase the operation count.”<sup>3</sup> (Reported by Max Hailperin.)

## Chapter 6

Page 95, the definition of *limit* of a range: the brackets should be hollow:

An iterator `f + n` is the limit of a half-open weak range  $\llbracket f, n \rrbracket$ .

(Reported by Eugene Kirpichov.)

## Chapter 8

Page 135, in line 9, “`t = successor(f)`” should be “`f = successor(t)`”. (Reported by Max Hailperin.)

## Chapter 12

Page 221, “performs reallocation only when `size(a) ≤ capacity(a)`” should be “performs reallocation only when the size after the insertion is greater than the capacity before the insertion”. (Reported by Max Hailperin.)

## Bibliography

Page 244, [Fiduccia 1985], change “*SIAM Journal of Computing*” to “*SIAM Journal on Computing*”.

Page 244, after [Fletcher and Silver 1966] insert “Floyd, Robert W. and Donald E. Knuth. 1990. Addition Machines. *SIAM Journal on Computing* 19(2): 329-0340.”

## Index

The following index items should all refer to page 40 rather than page 41:

page 250, second column: `binary_scale_down_nonnegative` and `binary_scale_up_nonnegative`  
page 253, first column: `even`  
page 253, second column: `half_nonnegative`  
page 255, second column: `negative`, `odd`, and `one`  
page 256, second column: `positive` and `predecessor`, of integer  
page 260, second column: `successor`, of integer  
page 261, first column: `twice`  
page 262, second column: `zero`

Also, on page 262, second column: the heading letter “X” should be “Z”.

## Code on web site

`type_functions.h`: The incorrect line:

```
#define NeedsDestruction(T) typename needs_construction_type<T>::type
```

has been corrected to:

```
#define NeedsDestruction(T) typename needs_destruction_type<T>::type
```

(Reported by Steven Gratton.)