Errata List for
MODULA-3 by Samuel P. Harbison
Last Modified: March 22, 1993

Modula-3 is the latest in the family of Pascal-like programming languages. It was jointly developed by DEC and Olivetti, and runs on many machines. Several features led us to adopt Modula-3 as a teaching vehicle in several courses. First, the Pascal-equivalent subset is easy to pick up. Second, explicit data abstraction makes learning data structure subjects much more meaningful. Third, its features make it an excellent object of study in programming language courses. Fourth, separation of specification from implementation, object orientation, and type inheritance make it a useful tool in software engineering. Fifth, threads and exceptions make it a good language for learning concepts in operating systems. Finally, it's free, so we can make it widely available, and you can obtain your own copy to do with as you please (more or less).

Currently, Sam Harbison's book, *MODULA-3*, is the standard tutorial and reference for learning and using the language. While we like this book a lot, there are many errors of various kinds in the text. So this document provides all the fixes we know about to make the book a reliable tutorial and reference.

The best approach is to systematically write in all the changes now, even before you've learned the contents of the sections. (The change 75:10 will require you to tuck in a new page, or format the change and tape it in.) You'll be using this book to learn Modula-3 incrementally in several courses, so the effort will be worthwhile.

Here are the errata in sequential order. Each entry has the form Page:Line:Change. Negative line numbers are counts from the bottom of the indicated page. This is based on changes required in the first printing; if you have a later printing, some of these errors may have been corrected. However, see 5:11 for an example of "re-bugging."

1:-4: Change "it easy" to "it is easy".

3:-6: Change "`MODULE Main`" to "`MODULE Hello1 EXPORTS Main`".

3:-1: Change "`END Main`" to "`END Hello1`".

5:11: The sentence should read, "Therefore, the characters `MODULEHello1` would be considered a single identifier rather than the reserved word `MODULE` followed by the identifier `Hello1`." This was correct in the first printing, but "`MODULEHello1`" may appear as "`MODULE Hello1`" in the second printing.

5:-5: Change "`MODULE Main`" to "`MODULE Hello1 EXPORTS Main`".

5:-4: Change "`END Main`" to "`END Hello1`".

8:7: Change "`Hello, World!`" to "`Hello, World!\n`".

8: : Swap positions of Example 1-4 and Figure 1-3.

9:20: Change “Expr END” to “Expr”.

10:-4: Rewrite last paragraph; mention left-associativity. Ask your instructor.

13:3: After this line add the line “Wr.Flush(Stdio.stdout);”.

20:3: Change “Threethousand” to “3thousand”.

20:-5: Mention that loop variable i is declared as READONLY in FOR statement.

20: : Swap positions of Box and Example 2-5.

23:-12: Change “i” to “k”.

31: : In 2.7.4, change “Fib” to “Fac” three times.

32:1: Add space before “By”.

32:2: Change “Fib” to “Fac”.

32:6: Change “n := n” to “n := BITSIZE(n)”.

32:7: Add “because the result type of BITSIZE is always CARDINAL.”

32:17: Change “is the type” to “is always the type”.

33:8 Wrong typeface.

33:15: Change “b:=c” to “b:INTEGER:=c”.

33:16: Change “c:=b” to “c:INTEGER:=b”.

33:-5: Add missing “)” at end of declaration.

33:-4: Add missing “)” at end of declaration.

35:1: Change “fifteen” to “eighteen”.

38:-17: Change “a,pos” to “a, pos”.

40:11: Change “Color are assignable” to “Color is assignable”.

40:-10: Change “Some the” to “Some of the”.

43:5: The box “Variations on Procedure Calls” makes an inaccurate statement about Modula-2. Modula-2 requires “()” in function calls but not in proper procedure calls.

43:-4: Change “assignable to N” to “assignable to n”.

47:-1: In box, change “Expr END” to “Expr”.

49:8: “Solution” should be boldface.

50:1: In box, keywords TO and DO should not be boldface.

57:8: Change "of record r" to "of record r and r is not READONLY".

57:-9: Rephrase as: "Which of the three versions is more readable?"

57:-5: Change "page 49" to "page 50".

64:-10: -8 MOD 3 is 1, not -1.

64:-4: x MOD 2 is 1 if x is odd; it cannot be -1.

67:19: Change "volume : [0..10]" to "volume: [0..10]". Squeeze out blank.

70:6: In the first line in the box, change "NTERFACE" to "INTERFACE".

72:7: In box, change "VALUE" to "VAL".

74:-3: Add: "and has type T" to example of FIRST and LAST.

74:-1: Add: "The type of FIRST and LAST is the base type of T. The type of NUMBER is CARDINAL."

75:10: The solution presented in Example 4-14 doesn't work when VolumeLevel is an enumeration or an integer subrange. The problem is that the value of NewVolume may lie outside the range of VolumeLevel and therefore cause a run-time error. Here is the new example and a revised following paragraph:

Example 4-14

Rewrite procedure ChangeVolume in Example 4-8 on page 67 assuming VolumeLevel is defined as the enumeration type Off, Soft, Medium, Loud, Ear_splitting.

Solution You cannot compute the tentative value of NewVolume as
VAL(ORD(volume)+n, VolumeLevel)
without getting a checked run-time error if n is too large or too small. This makes the procedure somewhat more cumbersome. Here is one possibility, in which the range testing is hidden in the functions MIN and MAX:

```
PROCEDURE ChangeVolume(n: INTEGER) =
  BEGIN
    WITH
      biggest = ORD(LAST(VolumeLevel)),
      smallest = ORD(FIRST(VolumeLevel)),
      proposed = ORD(volume) + n
    DO
      volume := VAL(MAX(MIN(biggest,proposed),smallest),VolumeLevel);
    END;
  END ChangeVolume;
```

Now suppose that v has the subrange type [T.A..T.B]. For the assignment v := e to be legal, the type of e must either be T or else it must be a subrange of T that includes at least one value in T.A..T.B. That is, the types of v and e must share at least one value. Even if the assignment is legal, it is a checked run-time error if the actual value of e is not in T.A..T.B.

84:7: In box, change "NTERFACE" to "INTERFACE".

90:6: Change "Ans" to "answer".

90:14: Change "character in t" to "character in str".

90:-10: Change "3. If t" to "3. If str".

111:6: Change "Point, the" to "Point (page 107), the".

111:-6: Change "SET OF t" to "SET OF T".

112:-5: Change "CARDINALITY" to "Cardinality".

112:-2: Identifiers in italics should be in the program typeface.

113:3: Change ":=" to "=".

124:6: Change "Sum" to "Add".

126:12: p and q are in the wrong typeface.

127:-2: Change "could passed" to "could be passed".

129:1: Remove "[A]".

129:-10: Change "the procedure" to "ForAllElements".

129:-3: Add "(or LAST(CARDINAL) if the array is empty)" before ":".

134:-18: Change "out.If" to "out. If".

136:1: In the box, the keyword ELSE should not be boldface.

141:18: Change "in procedure Forever" to "in the procedure".

141:-14: Change "ELSE =>" to "ELSE".

141:-9: Change "Chapter 12 (Threads)" to "Chapter 12".

141:-1: Change "exception" to "exceptions".

142:4: Change "Divsr" to "divisor".

146:9: In the caption for Figure 8-1, change "ServerImpl" to "Server".

153:13: Change "MODULE SetUnion" to "MODULE SetModule".

153:16: Add "BEGIN ..." before this line. Change "END SetUnion" to "END SetModule".

153:-16: Change "END P" to "END Empty".

155:-2: The last sentence should read: "Ordinary clients will import only Basic, while trusted clients will import both Basic and Friendly (Figure 8-5)."

168:1: In box, keyword END should not be boldface.

169:28: Insert line "BEGIN IntegerElement" before this line.

176:-13: Change "P" to "p" (five times) in Ex. 10-1 and following text.

177:-3: Change "beta := beta;" to "beta := alpha;".

178:13: Change "188" to "191".

178:-1: Add NEW to the list of built-in functions for reference types.

180:19: Add a semicolon after "theta := Math.Pi / 2.0".

181:-3: Change "number of names" to number of games".

184:6: Should read "The type REF T is a subtype of REFANY ...".

191:-19: Change "IMPORTS Rational" to "IMPORT Rational".

195:-7: Change "fill with gas" to "add gas".

196:-11: Change "type as opaque" to "type is opaque".

198:-17: Change "END SetBirthDate" to "END New".

199:-3: Change "B and E are" to "D and E are".

201:7: Change "Figure 11-2" to "Figure 11-4".

202:-16: Change "bigPoint" to "BigPoint".

210:-19: Change "page 196" to "page 197".

211:20: Change "variable a" to "variable square".

212:-4: Since $T_L = T_R$ implies $T_L <: T_R$, the first entry in the third row of the table is understood to mean $T_L <: T_R$ and $T_L \# T_R$. The table layout already suggests this interpretation.

214: : Add "RETURN self;" to InitPosition and InitRectangle.

214:-19: Editorial changes to paragraph "The Rectangle.init method...".

215:4: Change the second "ArticleCitation" to "ArticleTitle".

218: : Add "RETURN self;" to InitPosition and InitRectangle.

219: : Add "RETURN self;" to InitCircle.

219:-8: Editorial changes to the last paragraph.

221:4: Delete ":= NIL" from this line.

222:3: Insert "Wr.Flush(Stdio.stdout);" after this line.

222:-5: Change "METHODS" to "OVERRIDES".

226:-7: Change "Child" to "Child".

227:8: Mention that untraced things are discussed in 13.2 on p. 258.

227:-6: Change "name" to "name".

234: : Change "94681" to "94861" (four times).

244:24: Insert "RETURN Self;" before "END Init;".

271:12: Change "ELSEIF" to ELSIF".

280:12: In box, change "UnGetChar(rd: T; c: CHAR)" to "UnGetChar(rd: T)".

285:6: In box, insert line "PROCEDURE Bool(b: BOOL): Text.T;" after this line.

296:-7: Change "=QualID" to "= QualID".

299:3: Change "e. semantic" to "e. lexical or semantic".

299:5: The answer to D.1.5(c) is 11, not 3.

300:6: Change "Recs" to "recs".

300:8: Change "a := y" to "b := y".

300:-16: Change all "A" and "B" to "a" and "b" in answer to exercise 9.

300:-11: Delete procedure heading; the solution is a block statement.

305: : Correct typeface of leading entry in letter groups as necessary.

305: : Add "Alerted (Thread)" to the index (cite p. 250)

311:4: Change "Subtypes (see Types)" to "Subtypes (see Type)".

If you find more errors, please email the change to rro@cs.colostate.edu, or tell your instructor. We'll let you know when new versions of this document are available. Periodically, this is also announced in the comp.lang.modula3 newsgroup.