

## Lista de Erratas

Lamentavelmente, apesar de todo o esforço de escrita e revisão, o texto do livro contém erros. Esses erros serão corrigidos na próxima tiragem do livro. Segue abaixo uma lista dos erros já identificados:

1) Exemplo 2.13 (Página 36): há um espaço em branco após a vírgula que está sendo atribuída.

Como está no livro:

```
char virgula = ' , ';
```

Como deveria ser:

```
char virgula = ',';
```

2) Exemplo 2.21 (Página 41): não existe a palavra *then* associada ao primeiro comando *if* do exemplo.

Como está no livro:

```
if (n == 0) then {
```

Como deveria ser:

```
if (n == 0) {
```

3) Lista de Exercícios do Capítulo 2 (Página 44): há duas questões de número 6. O correto é que a segunda tenha o número 7.

4) Lista de Exercícios do Capítulo 2 (Página 44), na segunda questão de número 6: as funções *f* e *main* estão na ordem invertida, isto é o código da função *f* deveria estar antes do código da função *main*.

5) Linha com conversão binária para decimal (Página 48): o expoente do último número 2 deve ser 0 e não 1

Como está no livro:

$$11111101 = 1x2^7 + 1x2^6 + 1x2^5 + 1x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^1 = 253$$

Como deveria ser:

$$11111101 = 1x2^7 + 1x2^6 + 1x2^5 + 1x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 253$$

6) Linha com conversão binária para decimal (Página 49): o expoente do último número 2 deve ser 0 e não 1

Como está no livro:

$$00000011 = 0x2^7 + 0x2^6 + 0x2^5 + 0x2^4 + 0x2^3 + 0x2^2 + 1x2^1 + 1x2^1 = 3$$

Como deveria ser:

$$00000011 = 0x2^7 + 0x2^6 + 0x2^5 + 0x2^4 + 0x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 3$$

7) Figura 3.3 (Página 53): a sequência binária está incorreta, contendo valores de números superiores a um algarismo em algumas sequências de 4 dígitos binários

Como está no livro:

0000 0010	1010 0011	1011 0000	1110 0110	1111 1000	1000 1011
4 bytes sinal      7 casas inteiras 1 sinal				2 bytes 4 casas decimais	

Como deveria ser:

0000 0010	0010 0011	0011 0000	1000 0110	0111 1001	1000 0011
4 bytes sinal      7 casas inteiras 1 sinal				2 bytes 4 casas decimais	

8) Exemplo 3.2 (Página 57): falta a letra *m* de *medição* em uma das linhas do exemplo

Como está no livro:

*edicao.centimetros=180;*

Como deveria ser:

*medicao.centimetros=180;*

9) Figura 3.8 (Página 59): a segunda célula contém apenas 15 bits quando deveriam ser 16

10) Exemplo 3.15 (Página 74): uso da palavra *desreferenciamento* em vez de *derreferenciamento* na legenda do exemplo

Como está no livro:

*Exemplo 3.15 Desreferenciamento implícito em FORTRAN 90 e explícito em C*

Como deveria ser:

*Exemplo 3.15 Derreferenciamento implícito em FORTRAN 90 e explícito em C*

11) Exercício 6 do Capítulo 3 (Página 79): falta ponto e vírgula após a definição de *union cidadania* e *struct pessoa*.

Como está no livro:

```
union cidadania {  
    enum classe c;  
    enum instrucao i;  
}
```

Como deveria ser:

```
union cidadania {
    enum classe c;
    enum instrucao i;
};
```

Como está no livro:

```
struct amostra {
    int n;
    struct pessoa p[10];
}
```

Como deveria ser:

```
struct amostra {
    int n;
    struct pessoa p[10];
};
```

12) Exercício 7 do Capítulo 3 (Página 80): a linha *return p*; está com problema de indentação.

Como está no livro:

```
#include <iostream>
int& xpto (int sinal) {
    int p = 4;
    if (!sinal) {
        p*=sinal;
    } else {
        p++;
    }
    return p;
}
```

Como deveria ser:

```
#include <iostream>
int& xpto (int sinal) {
    int p = 4;
    if (!sinal) {
        p*=sinal;
    } else {
        p++;
    }
    return p;
}
```

13) Exercício 9 do Capítulo 4 (Página 104): falta *end if*; do comando *if* na *procedure B*.

Como está no livro:

```
procedure B (k: boolean);
  w: integer;
  begin -- B
    if k then
      B (false);
    else -- #;
  end B;
```

Como deveria ser:

```
procedure B (k: boolean);
  w: integer;
  begin -- B
    if k then
      B (false);
    else -- #;
    end if;
  end B;
```

14) Exemplo 4.11 (Página 99): existe uma linha adicional desnecessária com );

Como está no livro:

```
ResultSet resultados = comando.executeQuery (
  "SELECT nome, idade, nascimento FROM pessoa " +
  "WHERE idade < " + idadeMaxima);
);
while (resultados.next()) {
```

Como deveria ser:

```
ResultSet resultados = comando.executeQuery (
  "SELECT nome, idade, nascimento FROM pessoa " +
  "WHERE idade < " + idadeMaxima);
while (resultados.next()) {
```

15) Linha com exemplos de expressões literais (Página 107): a expressão literal “c” está com aspas duplas e deveria ser aspas simples

Como está no livro:

```
2.72 99 0143 “c” 0x63
```

Como deveria ser:

```
2.72 99 0143 ‘c’ 0x63
```

16) Linha em código ADA (Página 111): falta *end if*; ao final da linha

Como está no livro:

```
if x > y then max := x; else max := y;
```

Como deveria ser:

```
if x > y then max := x; else max := y; end if;
```

17) Tabela 5.2 (Página 113): As assinaturas da aplicação do operador \* sobre os tipos *char* e *short* deveriam retornar *int* por causa da promoção de tipos. Além disso, falta a assinatura da aplicação do operador \* sobre o tipo *byte* (*byte x byte* → *int*)

18) Exemplo 5.14 (Página 117): não existe a palavra *then* associada aos dois comandos *if* do exemplo

Como está no livro:

```
if (p instanceof Medico) then
    System.out.println ("Registre-se no CRM");
if (p instanceof Engenheiro) then
    System.out.println ("Registre-se no CREA");
```

Como deveria ser:

```
if (p instanceof Medico)
    System.out.println ("Registre-se no CRM");
if (p instanceof Engenheiro)
    System.out.println ("Registre-se no CREA");
```

19) Tabela 5.4 (Página 119): o operador de desigualdade da linha correspondente aos operadores de precedência de ordem 7 está incorreto (falta um !). Em vez de == =, a célula da tabela deveria conter == !=

20) Exemplo 5.17 (Página 120): não existe a palavra *then* associada ao comando *if* do exemplo

Como está no livro:

```
if (!x) then y = 3;
```

Como deveria ser:

```
if (!x) y = 3;
```

21) Linha de código em C (Página 123): o ; se encontra depois do } em vez de antes.

Como está no livro:

```
if (b < 2*c || a[i++] > c) { a[i++]++ };
```

Como deveria ser:

```
if (b < 2*c || a[i++] > c) { a[i]++; }
```

22) Linha de código em C com exemplos de atribuição unária (Página 125): falta ponto e vírgula após -- a.

Como está no livro:

```
+=a;          a++;          --a          a--;
```

Como deveria ser:

```
+=a;      a++;      --a;      a--;
```

23) Exercício 6 do Capítulo 5 (Página 140): o operador de negação - foi substituído indevidamente pelo operador de decremento - - nos comandos do exemplo

Como está no livro:

```
n = 3;
a = --n++;
a = --n+1;
a = --n+=1;
```

Como deveria ser:

```
n = 3;
a = -n++;
a = -n+1;
a = -n+=1;
```

24) Exemplo 6.16 (Página 165): falta <> envolvendo *math.h*

Como está no livro:

```
#include math.h
```

Como deveria ser:

```
#include <math.h>
```

25) Exemplo 6.16 (Página 165): o código da expressão retornada pela função *origem* está incorreto

Como está no livro:

```
return c.x*c.y*c.z;
```

Como deveria ser:

```
return c.x == 0 && c.y == 0 && c.z == 0;
```

26) Exemplo 6.22 (Página 172): *end* da função *ObtemTopo* se refere a *topo* e não a *ObtemTopo* como seria correto

Como está no livro:

```
end topo;
```

Como deveria ser:

```
end ObtemTopo;
```

27) Exemplo 6.23 (Página 175): falta retornar *void* no cabeçalho da função *tPilha::empilha (int el)*

Como está no livro:

*tPilha::empilha (int el) {*

Como deveria ser:

*void tPilha::empilha (int el) {*

28) Exemplo 6.23 (Página 175): cabeçalho da função *tPilha::obtemTopo (void)* está errado

Como está no livro:

*void tPilha::int obtemTopo (void){*

Como deveria ser:

*int tPilha::obtemTopo (void){*

29) Seção 6.2.3 – 3º parágrafo (Página 178): erro de tipografia na frase “Elas se diferenciam pelo forma como são usadas”.

Como está no livro:

Elas se diferenciam pelo forma como são usadas.

Como deveria ser:

Elas se diferenciam pela forma como são usadas.

30) Exemplo 7.3 (Página 197): Faltam aspas simples antes das listas nas chamadas da função *segundo*

Como está no livro:

*(segundo (1 2 3))*

*(segundo ( (1 2 3) (4 5 6)))*

*(segundo (“manga” “abacaxi” 5 6))*

Como deveria ser:

*(segundo ‘(1 2 3))*

*(segundo ‘( ‘(1 2 3) ‘(4 5 6)))*

*(segundo ‘(“manga” “abacaxi” 5 6))*

31) Figura 7.1 (Página 199): erro de indentação na quinta linha da figura.

Como está no livro:

Se T e T’ são primitivos, então T e T’ devem ser idênticos

Por exemplo, inteiro  $\equiv$  inteiro

Se T e T’ são produtos cartesianos e  $T = A \times B$  e  $T' = A' \times B'$ ,

então  $A \equiv A'$  e  $B \equiv B'$

Por exemplo, inteiro  $\times$  booleano  $\equiv$  inteiro  $\times$  booleano

Como deveria ser:

Se T e T’ são primitivos, então T e T’ devem ser idênticos

Por exemplo, inteiro  $\equiv$  inteiro

Se T e T’ são produtos cartesianos e  $T = A \times B$  e  $T' = A' \times B'$ ,

então  $A \equiv A'$  e  $B \equiv B'$

Por exemplo, inteiro x booleano  $\equiv$  inteiro x booleano

32) Quarta linha (Página 201): falta acento no u de reuso

33) Exemplo 7.23 (Página 218): Faltam dois fecho-chaves ao final do exemplo.

Como está no livro:

```
public class Empresa {
    public void paga (Pessoa pes) {}
    public void contrata (Empregado emp) {}
    public static void main(String[] args) {
        Pessoa p = new Pessoa ("Lucas", 30);
        Empregado e = new Empregado ("Luis", 23, 1500.00);
        p = e;
        // e = p;
        Empresa c = new Empresa();
        c.paga(e);
        // c.contrata(p);
    }
}
```

Como deveria ser:

```
public class Empresa {
    public void paga (Pessoa pes) {}
    public void contrata (Empregado emp) {}
    public static void main(String[] args) {
        Pessoa p = new Pessoa ("Lucas", 30);
        Empregado e = new Empregado ("Luis", 23, 1500.00);
        p = e;
        // e = p;
        Empresa c = new Empresa();
        c.paga(e);
        // c.contrata(p);
    }
}
```

34) Exemplo 7.31 (Página 224): Faltam abre e fecho parênteses após função *imprime*.

Como está no livro:

```
void imprime { cout << "Militar"; }
```

Como deveria ser:

```
void imprime( ) { cout << "Militar"; }
```

35) Exemplo 7.35 (Página 228): Existe um ponto e vírgula desnecessário ao final do método *temVirtual* do exemplo.

Como está no livro:

```
virtual void temVirtual ( ) { };
```

Como deveria ser:

```
virtual void temVirtual ( ) { }
```

36) Exemplo 7.36 (Página 231): Existem vários ponto e vírgulas desnecessários ao final da definição de todos os métodos do exemplo.

Como está no livro:

```
class Professor {  
    String n = "Marcos";  
    int matr = 53023;  
    public String nome() { return n; };  
    public int matricula() { return matr; };  
}  
class Aluno {  
    String n = "Marcos";  
    int matr = 127890023;  
    float coef = 8.3;  
    public String nome() { return n; };  
    public int matricula() { return matr; };  
    public float coeficiente() { return coef; };  
}  
class ProfessorAluno extends Professor {  
    Aluno aluno = new Aluno();  
    public float coeficiente() { return aluno.coeficiente(); };  
    public String matriculaAluno() { return aluno.matricula(); };  
}
```

Como deveria ser:

```
class Professor {  
    String n = "Marcos";  
    int matr = 53023;  
    public String nome() { return n; }  
    public int matricula() { return matr; }  
}  
class Aluno {  
    String n = "Marcos";  
    int matr = 127890023;  
    float coef = 8.3;  
    public String nome() { return n; }  
    public int matricula() { return matr; }  
    public float coeficiente() { return coef; }  
}  
class ProfessorAluno extends Professor {  
    Aluno aluno = new Aluno();  
    public float coeficiente() { return aluno.coeficiente(); }  
    public String matriculaAluno() { return aluno.matricula(); }  
}
```

37) Exemplo 7.36 (Página 231): método *matriculaAluno()* da classe *ProfessorAluno* deveria retornar *int* e não *String*.

Como está no livro:

```
public String matriculaAluno(){return aluno.matricula();};
```

Como deveria ser:

```
public int matriculaAluno(){return aluno.matricula();}
```

38) Exemplo 7.39 (Página 234): método *coeficiente()* da classe *Aluno* deveria retornar *float* e não *int*.

Como está no livro:

```
int coeficiente ( ) { return coef; }
```

Como deveria ser:

```
float coeficiente ( ) { return coef; }
```

39) Exemplo 8.2 (Página 248): Existe um ponto e vírgula desnecessário ao final da definição da função *executaFuncionalidade* do exemplo.

Como está no livro:

```
void executaFuncionalidade (int x ) {  
    printf("Faz alguma coisa!!!");  
};
```

Como deveria ser:

```
void executaFuncionalidade (int x ) {  
    printf("Faz alguma coisa!!!");  
}
```

40) Exemplo 8.2 (Página 248): Falta *()* na chamada de *trata1* e *trata2*.

Como está no livro:

```
void f(int x) {  
    if (condicao1(x)) trata1;  
    if (condicao2(x)) trata2;
```

Como deveria ser:

```
void f(int x) {  
    if (condicao1(x)) trata1();  
    if (condicao2(x)) trata2();
```

41) Exemplo 8.3 (Página 248): Falta *()* na chamada de *trata1*, *trata2* e *trata3*.

Como está no livro:

```
void g() {  
    int resp;  
    resp = f(7);
```

```
if (resp == 1) trata1;
if (resp == 2) trata2;
if (resp == 3) trata3;
```

Como deveria ser:

```
void g() {
    int resp;
    resp = f(7);
    if (resp == 1) trata1();
    if (resp == 2) trata2();
    if (resp == 3) trata3();
}
```

42) Exemplo 8.3 (Página 248): Falta *return 0*; ao final do código da função *f*.

Como está no livro:

```
int f(int x) {
    if (condicao1(x)) return 1;
    if (condicao2(x)) return 2;
    if (condicao3(x)) return 3;
    executaFuncionalidade(x);
}
```

Como deveria ser:

```
int f(int x) {
    if (condicao1(x)) return 1;
    if (condicao2(x)) return 2;
    if (condicao3(x)) return 3;
    executaFuncionalidade(x);
    return 0;
}
```

43) Exemplo 9.7 (Página 295): Existe um *\*/* desnecessário na penúltima linha

Como está no livro:

```
wait(&estado_filho); // aguarda o término do outro filho*/
```

Como deveria ser:

```
wait(&estado_filho); // aguarda o término do outro filho
```

44) Exemplo 9.8 (Página 297): Faltou um *s* em *lpthreads* na terceira linha do exemplo.

Como está no livro:

```
/* ex1.c */
/* Para compilar com gcc, use: */
/* gcc -lpthread -oex1 ex1.c */
```

Como deveria ser:

```
/* ex1.c */
/* Para compilar com gcc, use: */
```



```

Barrichelo----->
Barrichelo----->
Schumacher----->
Barrichelo----->
Schumacher----->
Barrichelo----->
Schumacher----->
Barrichelo----->
Schumacher----->
Barrichelo----->
Schumacher----->
Barrichelo----->
Barrichelo----->
Barrichelo----->
Barrichelo completou a prova.
Schumacher----->
Schumacher----->
Schumacher completou a prova.

```

47) Exercício 4 do Capítulo 9 (Página 314): Trocar “em substituição” por “junto” no enunciado.

Como está no livro:

Mostre como é possível utilizar semáforos em substituição aos laços *while*

Como deveria ser:

Mostre como é possível utilizar semáforos junto aos laços *while*

48) Exercício 7 do Capítulo 9 (Página 314): Trocar Exemplo 9.14 por Exemplo 9.13.

Como está no livro:

Suponha que sejam retiradas as chamadas às entradas *iniciar* de *carro1* e *carro2* no Exemplo 9.14.

Como deveria ser:

Suponha que sejam retiradas as chamadas às entradas *iniciar* de *carro1* e *carro2* no Exemplo 9.13.

49) Exercício 8 do Capítulo 9 (Página 315): Trocar Exemplo 9.10 por Exemplo 9.9.

Como está no livro:

Implemente o programa do Exemplo 9.10 retirando o semáforo

Como deveria ser:

Implemente o programa do Exemplo 9.9 retirando o semáforo

50) Último parágrafo do Prefácio (Página XII): Faltou especificar o site da Campus.

Como está no livro:

página do livro na Internet (acessível a partir de ).

Como deveria ser:

página do livro na Internet (acessível a partir de <http://www.campus.com.br>).