See also Generic class
example, 12-4

U

Unordered sequence classes
See also List class, Pair class, Association class, Hash_Table class; Ordered sequence classes; Parameterized templates
overview, 1-5
requirements for parameterized type, 7-1
use of operator=, 7-1

V

Vector class
base classes, 6-2
constructors, 6-2—6-3
description, 6-2
example, 6-8
friend functions, 6-7
member functions, 6-3—6-7
requirements for parameterized type, 6-1
use of operator=, 6-1
VERIFY macro
See also EXCEPTION macro; RAISE macro; STOP macro; IGNORE_ERRORS macro
description, 13-14
example, 13-15
group names as symbols, 13-14
synopsis, 13-15

  constructors, 2-11
  description, 2-11
  example, 2-12—2-14
  member functions, 2-11

S

Set class
  See also Hash_Table class; List class; Bit_Set class; Parameterized templates
  base classes, 8-2
  description, 8-2
  example, 8-5—8-6
  friend functions, 8-5
  member functions, 8-2—8-5
  public constructors, 8-2
  requirements for parameterized type, 8-1
  use of operator=, 8-1
Stack class
  base classes, 6-9
  constructors, 6-9
  description, 6-9
  example, 6-12
  friend functions, 6-11
  member functions, 6-9—6-11
  requirements for parameterized class, 6-1
  use of operator=, 6-1
STOP macro
  See also EXCEPTION macro; RAISE macro; VERIFY macro; IGNORE_ERRORS macro
  description, 13-13
  example, 13-14
  group names as symbols, 13-13
  synopsis, 13-13
String class
  See also Gen_String class; char* functions; Regexp class
  base class, 2-2
  constructors, 2-2
  definition, 2-1
  example, 2-7—2-8
  friend functions, 2-5—2-7
  member functions, 2-2—2-5
  operator char*, 2-1
  overview, 1-4
Stroustrup, Bjarne, 1-2, 10-1
SYM package
  See also symbol_package macro
  description, 11-21
  example of interface to, 11-21
  interface with, 11-21—11-22
Symbol class
  See also Package class; Generic class
  base class, 11-2
  definition of a symbol, 1-3, 11-1
  definition of property list, 11-1
  description, 11-2, 14-4
  friend class, 11-2
  friend functions, 11-3
  member functions, 11-3
  overview, 1-7, 14-4
  protected constructors, 11-2
  public constructors, 11-2
  use of operator=, 11-1
symbol_package macro
  See also DEFPACKAGE macro; DEFPACKAGE_SYMBOL macro
  contents of symbol package file, 11-20
  creation of SYM symbol package, 11-18
  creation of storage file, 11-18
  definition of additional macros, 11-18
  description, 11-17
  example, 11-18—11-20
  implementation details, 11-22—11-24
  implementation of package in symbols.C, 11-20
  synopsis, 11-18
Symbolic computing
  See also Symbol class; Package class
  COOL capabilities, 1-3
System interface classes
  See also Date_Time class; Timer class
  overview, 1-5

T

template
  See also Parameterized templates; container class
  syntax, 5-3
text_package macro
  adding translations for other languates, 11-17
  contents of symbol package file, 11-15
  creation of storage file, 11-12
  description, 11-12—11-13
  example, 11-13—11-17
  friend functions, 11-13
  implementation of package in symbols.C, 11-15
  support for multiple language translations, 11-13
  synopsis, 11-13
Time zone.h File
  See also Date_Time class
  description, 4-6
Timer class
  accuracy, 4-10
  constructors, 4-10
  description, 4-10
  example, 4-11
  member functions, 4-10
Tree classes
  See also Binary_Tree class; AVL_Tree class; N_Tree class; Node classes
  use of operator=, 9-1
TYPE_CASE macro
  See also Generic class
  description, 12-5
  example, 12-5
type_of

public constructors, 9-16
requirements for parameterized type, 9-1
traversal modes, 9-15
use of operator=, 9-1
Node and tree classes
See also Ordered sequence classes; Unordered sequence classes
overview, 1-6
Node classes. *See* Binary_Node class; N_Node class; D_Node class
Number classes
See also Random class; Complex class; Rational class; Bignum class; Range class
definition, 3-1
overview, 1-5

O

ONCE_ONLY macro
See also MACRO; once_only package
description, 10-9
example, 10-9
synopsis, 10-9
once_only package, description. *See* ONCE_ONLY macro
Ordered sequence classes
See also Parameterized templates; Unordered sequence classes; Vector class; Stack class; Queue class; Matrix class
overview, 1-5
requirements for parameterized type, 6-1
use of operator=, 6-1

P

Package class
See also Symbol class; Generic class; Macro
base classes, 11-4
constructors, 11-4
definition of a package, 1-3, 11-1
description, 11-3, 14-4
friend class, 11-4
friend functions, 11-6—11-7
member functions, 11-4—11-6
overview, 1-7
use of operator=, 11-1
Pair class
See also Association class
base class, 7-10
constructors, 7-10—7-11
description, 7-10
friend functions, 7-11
member functions, 7-11
use of operator=, 7-1
Parameterized classes. *See* Parameterized templates
Parameterized templates
See also Container classes
COOL, list of, 5-2
container class, description, 14-3
definition, 1-2
description, 5-1—5-3, 14-3—14-4
example of DECLARE and IMPLEMENT, 5-5
template example, 5-5—5-7
use of DECLARE and IMPLEMENT, 5-4
Polymorphic management
description, 12-1, 14-5
relationship with Generic and class macro, 14-5—14-6

Q

Queue class
base classes, 6-13
constructors, 6-13
description, 6-13
example, 6-16
friend functions, 6-16
member functions, 6-13—6-16
requirements for parameterized type, 6-1
use of operator=, 6-1

R

RAISE macro
See also EXCEPTION macro; STOP macro; VERIFY macro; IGNORE_ERRORS macro
description, 13-11—13-12
example, 13-12—13-13
group names as symbols, 13-12
synopsis, 13-12
Random class
base class, 3-2
Constructor, 3-2
description, 3-1
example, 3-3
member functions, 3-2
ONE_CONGRUENTIAL random generator, 3-2
SHUFFLE random generator, 3-2
SIMPLE random generator, 3-2
SUBTRACTIVE (Knuth) random generator, 3-2
THREE_CONGRUENTIAL random generator, 3-2
Range class
base class, 3-20
constructors, 3-20
description, 3-20
example, 3-21—3-22
member functions, 3-20—3-21
Rational class
base class, 3-9
constructors, 3-9
description, 3-9
example, 3-13
friend functions, 3-12
member functions, 3-9—3-11
Regexp class
See also Gen_String class
base class, 2-11

- description, 7-17
- example, 7-20—7-22
- friend functions, 7-20
- member functions, 7-17—7-20
- requirements for parameterized type, 7-1
- use of operator=, 7-1

heterogeneous containers
- *See also* Generic class
- example, 12-5—12-7

I

IGNORE macro
- *See also* MACRO
- description, 10-11
- example, 10-11
- synopsis, 10-11

IGNORE_ERRORS macro
- *See also* EXCEPTION macro; RAISE macro; STOP macro; VERIFY macro
- description, 13-17
- destructors not called, 13-17
- example, 13-18
- synopsis, 13-17

IMPLEMENT macro
- *See also* CCC; DECLARE macro; Parameterized templates
- description, 5-4
- example, 5-5
- synopsis, 5-4

INITIALIZE macro
- *See also* MACRO
- description, 10-10
- example, 10-11
- synopsis, 10-10

is_type_of
- *See also* Generic class
- example, 12-4

ISSAME macro
- *See also* MACRO
- description, 10-7
- example, 10-7
- synopsis, 10-7

Iterator class
- base class, 5-12
- constructors, 5-12
- description, 5-12
- example, 5-13
- member functions, 5-13
- provision of multiple iterators, 1-2
- state information, 5-12

J

Jump_Handler class
- *See also* Excp_Handler class; IGNORE_ERRORS macro
- base class, 13-16
- constructors, 13-16
- description, 13-16
- friend functions, 13-16
- group names as symbols, 13-16
- public methods, 13-16

K

KEYARGS macro
- *See also* MACRO
- description, 10-8
- example, 10-8
- synopsis, 10-8

L

List class
- base classes, 7-2
- constructors, 7-2
- description, 7-2
- example, 7-9—7-10
- friend functions, 7-8
- member functions, 7-2—7-8
- requirements for parameterized type, 7-1
- use of operator=, 7-1

M

MACRO
- description, 10-4
- examples, 10-5—10-7
- implementation, 14-2
- synopsis, 10-4

Matrix class
- base classes, 6-17
- constructors, 6-17
- description, 6-17
- example, 6-19
- friend functions, 6-19
- member functions, 6-17—6-19
- requirements for parameterized type, 6-1
- use of operator=, 6-1

misc.h, fundamental COOL constants, 1-4

N

N_Node class
- *See also* D_Node class; N_Tree class
- base class, 9-11
- description, 9-11
- friend classes, 9-11
- member functions, 9-11—9-12
- public constructors, 9-11
- requirements for parameterized type, 9-1

N_Tree class
- *See also* Binary_Tree class; AVL_Tree class; D_Node class; N_Node class
- base class, 9-16
- description, 9-15
- example, 9-18—9-19
- member functions, 9-16—9-17

description, 11-9
synopsis, 11-9

E

enumeration_package macro
creation of storage file, 11-11
description, 11-11
example, 11-11—11-12
synopsis, 11-11
use as dynamic enumeration types, 11-11
ERR_MSG text package
See also text package macro
creation of, 11-12
error messages in exceptions, 13-9, 13-18
Exception class, 13-7
as symbols in a package, 13-18
base class, 13-3
constructors, 13-3
description, 13-3—13-4
Error, 13-7
Error, default handler, 13-7
Fatal, default handler, 13-7
friend functions, 13-4
predefined types, 13-7
public data members in, 13-19
public methods, 13-4
System_Error, 13-7
System_Error, default handler, 13-7
System_Signal, 13-7
System_Signal, default handler, 13-8
Verify_Error, 13-7
Warning, 13-7
Warning, default handler, 13-8
Exception handling
See also Excp_Handler class
definition, 1-4
description, 13-1—13-2, 14-7
macros, 14-7
overview, 1-4
EXCEPTION macro
See also RAISE macro; STOP macro; VERIFY macro; IGNORE_ERRORS macro
description, 13-8
examples, 13-9—13-11
group names as symbols, 13-8
synopsis, 13-8
Exceptions
See also MACRO; Excp_Handler class; Symbol class; Package class
description, 13-1—13-2
description of COOL macros, 13-2
group names (aliases), 13-2
group names as symbols, 13-3
group names, example of, 13-9
overview, 14-6—14-7
predefined types, description, 14-7
public data members in user–defined exceptions, 13-19
user–defined types, 13-19
Excp_Handler class
See also Exception class; MACRO; Symbol class; Package class
as symbols in a package, 13-18
base class, 13-5
constructors, 13-5
dealing with exceptions, 13-5
description, 13-5
example, 13-6
friend class, 13-5
global exception handler stack, 13-2
group names, example of, 13-6
predefined types, 13-7
public methods, 13-5
EXPAND_ARGS macro
See also MACRO
description, 10-10
example, 10-10
synopsis, 10-10

G

Gen_String class
See also Regexp class; char* functions; String class
base class, 2-14
constructors, 2-14
definition, 2-14
example, 2-20
friend functions, 2-18—2-20
member functions, 2-14—2-18
operator char*, 2-14
Generic class
See also class macro; SYM package; Symbol class; Package class; class macro
addition of member functions, 12-2
base class, 12-2
definition, 1-3
description, 12-2, 14-5
example of runtime type checking. *See* Generic class
friend functions, 12-3
member functions, 12-3
overview, 1-7
protected constructors, 12-2
protected member functions, 12-2
relationship to SYM package, 12-2
symbols.C file, 12-2

H

Hash_Table class
base classes, 7-17
constructors, 7-17

  - data members, 14-8
  - description, 14-8—14-11
  - error message package, 14-8, 14-11
  - naming conventions
    - class, struct, typedef names, 14-9
    - constant declarations, 14-9
    - directory names, 14-9
    - function names, 14-9
    - global and static variables, 14-9
    - predicate function names, 14-9
    - preprocessor and macro names, 14-9
    - variable and data member names, 14-9
  - organization and contents of class header files, 14-8
  - private, protected, and public data, 14-10
    - regression test suite, 14-8, 14-11
  - source code documentation, 14-8
    - guidelines, 14-10—14-11
  - source code indentation, 14-11
  - source code layout, 14-8
  - source code system dependence, 14-8
  - source code system independence, 14-11
  - variable and class naming conventions, 14-8
- Complex class
  - base class, 3-4
  - constructors, 3-4
  - description, 3-4
  - example, 3-7—3-8
  - friend functions, 3-6—3-7
  - member functions, 3-4—3-6
- Container class
  - *See also* Parameterized templates; Ordered sequence classes; Unordered sequence classes
  - available in COOL, 5-10
  - current position, 5-11
  - definition, 1-2
  - description, 5-10
  - making your own, 5-13
  - member functions, 5-11
  - requirements for parameterized type, 5-14
  - storing objects in, 5-14
- COOL
  - audience, 1-1
  - class hierarchy, 1-7, 14-12
  - constants, defined in misc.h, 1-4
  - definition, 1-1
  - description of classes, 1-4—1-7
  - features, 1-1—1-7
  - introduction, 1-1
  - macros, 1-2
  - major features, 1-1
  - methodology, 14-1
  - porting to a new platform, 10-1
  - preprocessor
    - derived from, 1-2, 10-2
    - description, 10-1, 14-1—14-2
    - options, 10-2
    - synopsis, 10-2
  - symbolic computing capabilities, 14-4
  - TRUE and FALSE constants, 1-4
- Country.h File
  - *See also* Date_Time class
  - description, 4-7
- Current position
  - *See also* Iterator class
  - description of, 5-11
  - example, 10-6
  - member functions, 5-11
  - state information, 5-12

D

- D_Node class
  - *See also* N_Node class; N_Tree class
  - base class, 9-13
  - description, 9-13
  - friend classes, 9-13
  - member functions, 9-13—9-14
  - public constructors, 9-13
  - requirements for parameterized type, 9-1
- Date_Time class
  - base class, 4-1
  - constructors, 4-1—4-2
  - description, 4-1
  - example, 4-8—4-9
  - friend functions, 4-5—4-6
  - member functions, 4-2—4-5
- DECLARE macro
  - *See also* IMPLEMENT macro; Parameterized Templates
  - description, 5-4
  - example, 5-5
  - synopsis, 5-4
- defmacro
  - description, 10-3
  - implementation, 10-3
  - options, 10-3
  - synopsis, 10-3
- DEFPACKAGE macro
  - *See also* DEFPACKAGE_SYMBOL macro; MACRO; enumeration_package, text_package; symbol_package; once_only package
  - allocation of storage, 11-10
  - creating specialized packages, 11-10
  - definition, 1-3
  - description, 11-7
  - importance of symbols.C file, 11-10
  - options, 11-7—11-8
  - synopsis, 11-7
- DEFPACKAGE_SYMBOL macro
  - *See also* DEFPACKAGE macro
  - adding symbols to a package, 11-9

INDEX

Symbols

#pragma defmacro
- description, 10-1
- implementation. *See* defmacro
- overview, 1-2

A

Association class
- *See also* Pair class
- base classes, 7-12
- constructors, 7-12
- description, 7-12
- example, 7-15—7-17
- friend functions, 7-15
- member functions, 7-12—7-15
- requirements for parameterized type, 7-1
- use of operator=, 7-1

AVL_Tree class
- *See also* Binary_Tree class; N_Tree class
- base class, 9-7
- description, 9-6—9-7
- example, 9-9—9-10
- friend functions, 9-9
- member functions, 9-7—9-9
- public constructors, 9-7
- requirements for parameterized type, 9-1
- use of operator=, 9-1

B

Bignum class
- base class, 3-15
- constructors, 3-15
- description, 3-14
- example, 3-19
- friend functions, 3-17—3-18
- maximum value, 3-14
- member functions, 3-15—3-17
- parsing of character string representations, 3-14

Binary_Node class
- *See also* Binary_Tree class; N_Node class; D_Node class
- base class, 9-2
- constructors, 9-2
- description, 9-2
- friend class, 9-2
- member functions, 9-2
- requirements for parameterized type, 9-1

Binary_Tree class
- *See also* Binary_Node class; AVL_Tree class; N_Tree class
- base classes, 9-3
- description, 9-3
- example, 9-5—9-6
- friend functions, 9-5
- member functions, 9-3—9-5
- public constructors, 9-3
- requirements for parameterized type, 9-1
- use of operator=, 9-1

Bit_Set class
- *See also* Set class; enumeration_package
- base classes, 8-6
- description, 8-6
- example, 8-11
- friend functions, 8-10
- member functions, 8-6—8-10
- public constructors, 8-6

C

Calendar.h File
- *See also* Date_Time class
- description, 4-8

CCC
- *See also* IMPLEMENT macro; Parameterized templates
- definition, 1-2, 5-7
- description, 5-7—5-8, 14-4
- example, 5-9
- options, 5-8

char* functions
- *See also* String class; Gen_String class
- description, 2-8
- friend functions, 2-8—2-10

class macro
- *See also* Generic class; MACRO
- definition, 1-4
- description, 12-7, 14-6

classmac macro
- *See also* Generic class; MACRO
- arguments, 12-8
- description, 12-7, 14-6
- example, 12-9—12-10
- interaction with class macro, 12-7
- synopsis, 12-8

Coding style
- build procedure, 14-8, 14-11
- class header file organization, 14-9—14-10

Printed on: Wed Apr 18 07:16:33 1990

Last saved on: Tue Apr 17 12:10:37 1990

Document: index

For: skc