Appendix A

C++ Operators

<table>
<thead>
<tr>
<th>Description</th>
<th>Operator</th>
<th>Arity</th>
<th>Prec.</th>
<th>Assoc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>scope</td>
<td>::</td>
<td>2</td>
<td>18</td>
<td>right</td>
</tr>
<tr>
<td>subscript</td>
<td>[]</td>
<td>2</td>
<td>17</td>
<td>left</td>
</tr>
<tr>
<td>function call</td>
<td>()</td>
<td>2</td>
<td>17</td>
<td>left</td>
</tr>
<tr>
<td>construction</td>
<td>type()</td>
<td>1</td>
<td>17</td>
<td>right</td>
</tr>
<tr>
<td>member access</td>
<td>.</td>
<td>2</td>
<td>17</td>
<td>left</td>
</tr>
<tr>
<td>member access</td>
<td>-&gt;</td>
<td>2</td>
<td>17</td>
<td>left</td>
</tr>
<tr>
<td>post-increment</td>
<td>++</td>
<td>1</td>
<td>17</td>
<td>left</td>
</tr>
<tr>
<td>post-decrement</td>
<td>--</td>
<td>1</td>
<td>17</td>
<td>left</td>
</tr>
<tr>
<td>dynamic cast</td>
<td>dynamic_cast&lt;&gt;</td>
<td>1</td>
<td>17</td>
<td>right</td>
</tr>
<tr>
<td>static cast</td>
<td>static_cast&lt;&gt;</td>
<td>1</td>
<td>17</td>
<td>right</td>
</tr>
<tr>
<td>reinterpret cast</td>
<td>reinterpret_cast&lt;&gt;</td>
<td>1</td>
<td>17</td>
<td>right</td>
</tr>
<tr>
<td>const cast</td>
<td>const_cast&lt;&gt;</td>
<td>1</td>
<td>17</td>
<td>right</td>
</tr>
<tr>
<td>type Identification</td>
<td>typeid</td>
<td>1</td>
<td>17</td>
<td>right</td>
</tr>
<tr>
<td>pre-increment</td>
<td>++</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>pre-decrement</td>
<td>--</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>dereference</td>
<td>*</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>address</td>
<td>&amp;</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>bitwise complement</td>
<td>~</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>logical not</td>
<td>!</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>sign</td>
<td>+</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>sign</td>
<td>-</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>sizeof</td>
<td>sizeof</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>new</td>
<td>new</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>delete</td>
<td>delete</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>cast</td>
<td>(type)</td>
<td>1</td>
<td>16</td>
<td>right</td>
</tr>
<tr>
<td>member pointer</td>
<td>-&gt;*</td>
<td>2</td>
<td>15</td>
<td>left</td>
</tr>
<tr>
<td>member pointer</td>
<td>.*</td>
<td>2</td>
<td>15</td>
<td>left</td>
</tr>
</tbody>
</table>

Table 6: Precedences and associativities of C++ operators.
Index

*this, 191

access
  data member, 160
  member function, 191
access restrictions
  private, 190
  public, 190
access specifier
  private, 190
  public, 190
Ackermann function, 143
addition
  assignment operator, 48
  addition operator, 45
address
  of a memory cell, 14
  of a variable, 25
  of an object, 26
  alternate denial, 64
ANSIC
  random number generator, 205
antisymmetry, 64
application program, 8
argument-dependent look up, 164
Arithmetic Evaluation Rule 1, 40
Arithmetic Evaluation Rule 2, 40
Arithmetic Evaluation Rule 3, 43
arithmetic expression, 26
arithmetic operators, 45
arithmetic type, 39
arity, 29
ASCII code, 182
assertion, 194
assignment
member-wise, 161
  of a struct value, 161
  of reference, 170
assignment operator, 30
associative operation, 40
associativity
  left, 41
  right, 41
associativity of operator, 40
automatic storage duration, 78
base
  of a floating point number system, 103
binary expansion
  of natural number, 51
  of real number, 104
binary operator, 29
binary representation
  of int value, 53
  of unsigned int value, 52
binary-to-decimal conversion, 52
bit, 14
bitwise operators, 64
block, 74
BODMAS, 40
body
  of do statement, 81
  of for statement, 71
  of function, 120
  of while statement, 79
bool, 61
Boolean, 59
Boolean expression, 62
Boolean function, 59
INDEX

completeness, 60
break statement, 82
bug, 75
built-in type, 24
burst, 56
C++ standard, 22
call by reference, 171
call by value, 171
call parameters, 121
call stack, 138
cancellation
  in floating point computations, 111
cast (functional notation), 101
cast expression, 101
central processing unit (CPU), 14
char, 182
charter, 182
control, 182
choosing numbers
game, 203
Church, Alonso (1903–1995), 89
Church Turing thesis, 89
class
  access specifier, 200
  constructor, 193
definition, 199
  implementation, 200
  member declaration, 200
  member function, 190
  member function call, 192
  member operator, 197
  method, 192
  nested type, 198
  private member, 190
  public member, 190
class scope, 200
class type, 200
Collatz problem, 79
Collatz, Lothar (1910–1990), 90
comment, 22
compilation, 13
compile, 12
compiler, 13
composite expression, 26
compound statement, 74
computable function, 89
computer
  main memory, 14
  memory cell, 14
  processor, 14
  von Neumann, 14
condition
  of a while statement, 79
  of an if statement, 69
conditional operator, 90
const
  member function, 192
  Const Guideline, 181
  Const-qualified type, 178
cast-reference, 179
con-type, 178
class
  initialization, 178, 179
  constructor, 193
default, 195
  explicit call, 194
  initialiser, 193
continue statement, 83
control character, 182
control flow
  iteration, 70
  jump, 82
  loop, 69
  selection, 69
control statement, 70
casting
  explicit, 101
  floating point, 100
  implicit, 50
  integral, 50
  promotion, 63
  standard, 186
  user defined, 196
CPU, 14
data encapsulation, 189
data member
  access for, 160
  of struct, 159
  De Morgan’s laws, 62
decimal-to-binary conversion, 51
declaration, 32
debugging output, 75
dependent region, 76
default constructor, 195
default initialization
  struct, 161
debugging, 32
  by default constructor, 195
definition
  of a class, 199
  of a function, 120
  of a variable, 25
denormalized number, 114
directive
  include, 23
discarding constant, 179
discriminant
  of a quadratic equation, 111
division assignment operator, 48
do statement, 81
domain
  of a function, 119
double, 98
draw
  random number generator, 203
editor, 12
effect
  of a statement, 32
  of an expression, 26
effect (semantical term), 24
equality
  of data, 189
  Euclidean algorithm, 139
evaluation
  of an expression, 26
  short circuit, 64
evaluation sequence, 44
executable, 13
exception
  explicit conversion, 101

  exponent
    of a floating point number, 103
expression
  arithmetic, 26
  Boolean, 62
expression of, 26
expression sequence, 44
expression type, 121
expression, 43

  literal, 25
expression of,
  value, 28
  mixed, 50
  of type void, 122
  order of effects, 55
  primary, 26
  rvalue, 28
  type of, 26
  valid, 27
  value of, 26
  variable, 25
expression statement, 32
file, 15
finite floating point number system, 103
INDEX

secondary operand, 41
long double, 113
long int, 54
look up
argument-dependent, 164
loop, 70
infinite, 72
progress towards termination, 71
value, 28
value-to-value conversion, 29
Mac OS, 15
machine epsilon, 106
machine language, 12, 15
macro, 194
main function, 23
main memory, 14
mantissa
of a floating point number, 103
mathematical induction, 140
member access
in a struct, 160
member access operator, 160
member function, 190
access for, 191
call, 192
implicit call parameter, 191
member function
and const, 192
member operator
of class, 197
member specification
of a struct, 159
member-wise assignment, 161
member-wise initialization, 161
memory cell, 14
Mersenne primes, 9
method
of class, 192
mixed expression, 50
modularization, 124
mod plus assignment operator, 48
mod plus operator, 46
multiplication assignment operator, 48
multiplication operator, 30, 45
name
class, 23
hiding, 76
of a class, 200
of a function, 120
of a type, 24
of a variable, 25
of formal parameter, 120
qualified, 23
unqualified, 23
namespace, 23
namespace scope, 76
nested type, 198
normalized floating point number, 103
null statement, 32
numeric limits
of floating point types, 113
of integral types, 49
object, 26
object code, 126
open source software, 127
operand, 29
operating system (OS), 15
Lunix, 15
Mac OS, 15
Unix, 15
Windows, 15
operator
addition, 45
addition assignment, 48
arithmetic, 45
arithmetic assignment, 48
arity, 29
assignment, 30
associativity, 40
binary, 29
bitwise, 40
conditional, 90
division assignment, 48
function call, 202
functional notation, 163
functionality, 29
infix notation, 163
input, 30
integer division, 46
logical, 61
modulus, 46
modulus assignment, 48
multiplication, 30, 45
multiplication assignment, 48
operand, 29
output, 31
overloading, 162
post-decrement, 47
post-increment, 47
pre-decrement, 47
pre-increment, 47
precedence, 40
relational, 62
return value, 29
subtraction, 45
subtraction assignment, 48
ternary, 90
unary, 29
unary minus, 47
unary plus, 47
operator token, 29
overloaded, 42
order of effects, 55
OS, 15
output operator, 31
output stream, 31
overflow
of value range, 49
overloading
argument-dependent lookup, 164
best match, 165
of functions, 162
of operators, 162
overloading resolution, 164
PEMDAS, 40
platform, 15
point of declaration, 77, 89
porting, 12
post-decrement operator, 47
post-increment operator, 47
postcondition
of a function, 119
potential scope, 77
pre-decrement operator, 47
pre-increment operator, 47
precedence of operator, 40
precision
of a floating point number system, 103
precondition
of a function, 119
primary expression, 26
primitive recursion, 142
private
class member, 190
private, 190
processor, 14
production
of a Lindenmayer system, 144
program state, 14
programming language, 8
promotion, 63
bool to int, 63
char to int, 182
float to double, 100
pseudorandom numbers, 200
public
class member, 190
public, 190
qualified function call, 164
qualified name, 23
RAM, 14
random access memory (RAM), 14
random number, 200
random number generator, 200
ANSI, 205
INDEX

Sheffer stroke, 64  
Sheffer, Henry M. (1883–1964), 64  
short circuit evaluation, 64  
short int, 54  
side effect, 27  
sign  
of a floating point number, 103  
signature of a function, 132  
signed char, 54  
significant  
of a floating point number, 103  
Single Modification Rule, 55  
sourcecode, 13  
availability, 127  
special character, 182  
standard conversion, 196  
standard error, 31  
standard input, 30  
standard library, 23  
mathematical functions, 132  
std::cerr, 31  
std::cin, 30  
std::cout, 31  
std::sqrt, 130  
standard output, 31  
statement, 32  
break, 32  
compound, 74  
continue, 93  
control, 70  
declaration, 32  
do, 81  
extinction, 32  
extension, 32  
of, 74  
goto, 91  
if, 69  
ifelse, 70  
iteration, 70  
jump, 82  
null, 32  
return, 93, 91  
selection, 70  

INDEX

switch, 88  
while, 79  
static storage duration, 78  
static variable, 78, 91  
std:cerr, 31  
std::cin, 30  
std::cout, 31  
std::sqrt, 130  
storage duration, 78  
automatic, 78  
static, 78  

INDEX

struct, 157  
assignment, 161  
data member, 159  
declaration, 160  
default initialization, 161  
definition, 158  
functionality, 160  
n_initialization, 161  
member access, 160  
member specification, 159  
underlying type, 159  
value range, 159  
semantical, 161  
syntactical, 161  

INDEX

subtraction assignment operator, 48  
subtraction operator, 45  
switch statement, 88  
syntactical value range of a struct, 161  
syntax, 21  

INDEX

tail-end recursion, 141  
temporary object, 171  
termination, 140  
ternary operator, 90  
topological sorting, 44  
true, 61  

INDEX

turing machine, 89  
turing, Alan (1912–1954), 89  
turtle graphics, 145  
two's complement, 53  
type, 24  

INDEX

arithmetic, 39  
bool, 61  
built-in, 24  
char, 182  
class, 200  
const, 178  
constant qualified, 178  
double, 98  
float, 98  
floating point, 97  
fundamental, 24  
fundamental, 24  
integral, 53  
long double, 113  
long int, 54  
namer of, 24  
of a typedef, 25  
of an expression, 26  
of a formal parameter, 120  
reference, 170  
short int, 54  
signed char, 54  
underlying a struct, 159  
unsigned char, 55  
unsigned int, 50  
unsigned long int, 55  
unsigned short int, 55  

INDEX

value range of, 24  

INDEX

void, 121  
typedef, 199  

INDEX

unary minus operator, 47  
unary operator, 29  
unary plus operator, 47  
undecidable problem, 89  
undefined behavior, 21, 27  

INDEX

underflow  
of value range, 49  
Unix, 15  

INDEX

unqualified function call, 164  
unqualified name, 23
unsigned char, 55
unsigned int, 50
unsigned long int, 55
unsigned short int, 55
user-defined conversion, 196
using directive, 35

valid expression, 27
value
  of a variable, 25
  of an expression, 26
value (semantical term), 24
value range
  of a struct, 159
  of a type, 24
  of type bool, 61
  of type double, 107
  of type float, 107
  of type int, 48
  of type unsigned int, 50
overflow, 49
semantical, 161
syntactical, 161
underflow, 49
variable, 25
  address of, 25
  name of, 25
  static, 79, 91
  type of, 25
  value of, 25
variable declaration, 25
variable definition, 25
visibility
  of name, 77
void, 121
void function, 121
von Neumann computer, 14

while statement, 79
  body, 79
Windows, 15
zero initialization, 91