

# Princeton University

## COS 217: Introduction to Programming Systems

### C Operators

**Grouped by Category:**

| Operator | Precedence | Category   | Description                            | Associativity |
|----------|------------|------------|--|---------------|
| ++       | 2          | arithmetic | Increment                              | R to L        |
| --       | 2          | arithmetic | Decrement                              | R to L        |
| +        | 2          | arithmetic | Unary positive                         | R to L        |
| -        | 2          | arithmetic | Unary negative                         | R to L        |
| *        | 3          | arithmetic | Multiplication                         | L to R        |
| /        | 3          | arithmetic | Division                               | L to R        |
| %        | 3          | arithmetic | Modulus                                | L to R        |
| +        | 4          | arithmetic | Addition                               | L to R        |
| -        | 4          | arithmetic | Subtraction                            | L to R        |
| =        | 14         | assignment | Assignment                             | R to L        |
| +=       | 14         | assignment | Addition and assignment                | R to L        |
| -=       | 14         | assignment | Subtraction and assignment             | R to L        |
| *=       | 14         | assignment | Multiplication and assignment          | R to L        |
| /=       | 14         | assignment | Division and assignment                | R to L        |
| %=       | 14         | assignment | Modulus and assignment                 | R to L        |
| <        | 6          | relational | Less than                              | L to R        |
| <=       | 6          | relational | Less than or equal to                  | L to R        |
| >        | 6          | relational | Greater than                           | L to R        |
| >=       | 6          | relational | Greater than or equal to               | L to R        |
| ==       | 7          | relational | Equality                               | L to R        |
| !=       | 7          | relational | Inequality                             | L to R        |
| !        | 2          | logical    | Logical "not"                          | R to L        |
| &&       | 11         | logical    | Logical "and"                          | L to R        |
|          | 12         | logical    | Logical "or"                           | L to R        |
| []       | 1          | pointer    | Array element select                   | L to R        |
| *        | 2          | pointer    | Dereference                            | R to L        |
| &        | 2          | pointer    | Address of                             | R to L        |
| ->       | 1          | structure  | Structure dereference and field select | L to R        |
| .        | 1          | structure  | Structure field select                 | L to R        |
| ~        | 2          | bitwise    | Bitwise "not"                          | R to L        |
| <<       | 5          | bitwise    | Bitwise shift left                     | L to R        |
| >>       | 5          | bitwise    | Bitwise shift right                    | L to R        |
| &        | 8          | bitwise    | Bitwise "and"                          | L to R        |
| ^        | 9          | bitwise    | Bitwise "exclusive or"                 | L to R        |
|          | 10         | bitwise    | Bitwise "or"                           | L to R        |
| &=       | 14         | bitwise    | Bitwise "and" and assignment           | R to L        |
| ^=       | 14         | bitwise    | Bitwise "exclusive or" and assignment  | R to L        |
| =        | 14         | bitwise    | Bitwise "or" and assignment            | R to L        |
| <<=      | 14         | bitwise    | Bitwise left shift and assignment      | R to L        |
| >>=      | 14         | bitwise    | Bitwise right shift and assignment     | R to L        |
| ()       | 1          | function   | Function call                          | L to R        |
| (type)   | 2          | cast       | Cast                                   | R to L        |
| sizeof   | 2          | sizeof     | size of (completetime)                 | R to L        |
| ?:       | 13         | ternary    | Conditional expression (ternary)       | R to L        |
| ,        | 15         | sequence   | Sequence                               | L to R        |

## Differences between C and Java

### Java only:

|            |   |
|------------|---|
| >>>        | Right shift with zero extension                   |
| new        | Create an object                                  |
| instanceof | Is left operand an object of class right-operand? |

### C only:

|        |                         |
|--------|-------------------------|
| ->     | structure member select |
| *      | dereference             |
| &      | address of              |
| ,      | sequence                |
| sizeof | compiletime sizeof      |

### Related to type boolean:

|       |  |
|-------|--|
| Java: | Relational and logical operators evaluate to type <code>boolean</code> |
| C:    | Relational and logical operators evaluate to type <code>int</code>     |
| Java: | Logical operators take operands of type <code>boolean</code>           |
| C:    | Logical operators take operands of type <code>int</code>               |

### Related to class String:

|       |  |
|-------|--|
| Java: | Operators <code>+</code> and <code>+=</code> can concatenate <code>String</code> objects   |
| C:    | Operators <code>+</code> and <code>+=</code> do not concatenate <code>String</code> objects – because there are no <code>String</code> objects |

Java: Demotions are not automatic

C: Demotions are automatic

```
int i ;
char c;
...
i = c;          /* Implicit promotion. */
                /* OK in Java and C. */

c = i;          /* Implicit demotion. */
                /* Java: Compiletime error. */
                /* C: OK. Truncation without warning. */

c = (char)i;    /* Explicit demotion. */
                /* Java: Truncation without warning. */
                /* C: Truncation without warning. */
```

Copyright © 2015 by Robert M. Dondero, Jr.