DEFAULT ARGUMENTS

Not supported by Java

Delroy A. Brinkerhoff

DEFAULT ARGUMENTS

- Balance flexibility and ease of use
- The ignore function
  - `cin.ignore();`
  - `cin.ignore(10);`
  - `cin.ignore(10, ‘:’);`

DEFAULT ARGUMENTS VS. OVERLOADED FUNCTIONS

```
void window(int width, int height){...}
void window(int width, int height, int x, int y, int color){...}


void window(int width, int height, int x = 0, int y = 0, int color = WHITE){...}

window(100, 200);
window(100, 200, 50);
window(100, 200, 50, 75);
window(100, 200, 50, 75, BLUE);
```

DEFAULT ARGUMENTS: RULE 1

- No argument without a default may be defined to the right of an argument with a default.

```
void function(int a, int b = 1, int c = 2);   // okay
void function(int a, int b = 1, int c);       // error
```

DEFAULT ARGUMENTS: RULE 2

- When calling a function with default arguments, an argument may not be specified to the right of an argument whose default is accepted.

```
function(50);                 // okay
function(60, 70);             // okay
function(60, 70, 80);         // okay
function(60, , 80);           // error
```

DEFAULT ARGUMENTS: RULE 3

- If the function definition and the function declaration (i.e., the prototype) are not the same, then the default values appear in function prototypes.
- Function Prototype

```
void function(int a, int b = 1, int c = 2);
```

- Function Definition

```
void function(int a, int b, int c) { . . . }
```

DEFAULT ARGUMENTS: RULE 4
NO CONFLICT

- If a function with default arguments is also overloaded, all possible ways that the function can be called, with and without default values, must be distinct from any and all overloaded versions.

```
void function();                                 // function 1
void function(int a, int b = 1, int c = 2);      // function 2

function();                                      // function 1
function(10);                                    // function 2
function(10, 20);                                // function 2
function(10, 20, 30);                            // function 2
```

DEFAULT ARGUMENTS: RULE 4
AMBIGUOUS FUNCTION CALLS

```
void function(int a);                          // function 1
void function(int a, int b = 1, int c = 2);    // function 2

function(10);                                  // which one??
function(10, 20);                              // function 2
function(10, 20, 30);                          // function 2
```