



# From Java to C++

# Java and C++ are very similar

## ■ Similar in:

- Syntax: Java used syntax similar to C++ to ease adoption
- Principles: Both are object-oriented languages
- Execution: Many similarities when run on a machine
  - Compiled down to similar assembly language

## ■ Different in goals:

- Java designed for: safety and portability
- C++ designed for: performance and control

*As a result, C++ exposes aspects of execution that Java hides*

# Review: Stack and Heap

- Stack: automatic variables, local variables in functions
- Heap: all objects and arrays are on the heap

*reclaimed when leave scope*

# Call by value

- When parameters are passed, we make a copy
  - For primitives, we copy the primitive
  - For objects, we copy the reference (pointer)

a) 0      b) 1

e) mug coffee

# Java Arrays are objects

- Allocated on the heap

# C++ code doesn't need to be in classes

- All programs need a “main” function

```
int main() {  
    return 0;  
}
```

*error code* (pointing to main)

*NO ERROR* (pointing to `return 0;`)

```
int main(int argc, char *argv[])
```

*length array* (pointing to `argc`)

*array* (pointing to `argv`)

- Can write code directly in this main

# Namespaces

- **Allows different people to use the same names**
  - Similar to Java in principle
- Uses “scope resolution” operator `::`

# Structure of C++ class

- **Two parts: a form of encapsulation**
  - header file (something.h) provides interface
  - body (something.cpp) provides implementation
- **Standard structure:**
  - “include guards” in header file