

Review

1. What does the compiler do?
 - How is compiling different from interpreting?
2. What are examples of compiler optimizations?
3. True or False: If the compiler is applying lots of code optimizations to my code, that means I wrote my code poorly.
4. Compare Java and Python
 - First, focus on their characteristics (just the facts, not tradeoffs)
 - Then, think about pros and cons, preferences

Compiler Optimization: Example 5

```
class Parent {  
    void final f() {  
        System.out.println("f");  
    }  
}
```

```
for( Parent p : parentArray ) {  
    p.f();  
}
```

Optimization:

```
for( Parent p : parentArray ) {  
    System.out.println("f");  
}
```

Compiler Optimization: Example 5

```
class Parent {  
    void final f() {  
        System.out.println("f");  
    }  
}
```

```
for( Parent p : parentArray ) {  
    p.f(); // check p's actual type at runtime  
          // and execute its method f  
}
```

Optimization:

```
for( Parent p : parentArray ) {  
    System.out.println("f");  
}
```

Different Perspectives on the Program

To the Compiler

- This is my one shot to validate the program and optimize it!

To You/Developer: long view

- I am compiling the program now, but I could change the program later.
- It should be easy to update the program; otherwise, I could introduce bugs.

Compiler Tradeoffs

- Upfront costs
 - Searching for optimizations
 - Make optimizations
 - Typically not Big-O efficiency improvements (unless program is written really inefficiently)
 - Iterative process: compiler makes optimizations and then looks for more optimizations
- Improved runtime
 - Expect executed many more times than compiled

LANGUAGE COMPARISON

Language Comparison

Java

Python

- 1) Focus on their characteristics (just the facts, not tradeoffs)
- 2) Pros and cons, preferences

Language Comparison

Java

- Entirely Object-oriented*
 - Procedural
 - Functional - newer
- Statically, strongly typed
- Compiled

Python

- Object-oriented
 - Also procedural and functional programming
- Dynamically, strongly typed
- Interpreted

Pros and cons of using each?

Rest of the Semester

- Shift from learning Java, specifically, to learning how to develop software (abstractly) with Java as our implementation/example
- Why Java?
 - Popular language
 - Many frameworks and tools for Java
 - Java's structure allows for strict adherence to design techniques
- Just a start on Java
 - You'll need to continue learning more Java on your own

Looking Ahead

- Canvas “quiz” – before Wednesday’s class
 - Read slides about testing, JUnit
 - Answer questions in “quiz”
- Goal: Hands-on lab in class on Wednesday
 - Prep: Clone the repository