Program Representation

Programming Language

Overview and purpose of program representations, understanding Java byte code

Program Representation

SP-2021 Class Session 2

Questions about Videos?

Video 1: Introduction
Watch this video introducing program representations

Video 2: Source Code, Byte Code, Machine Code
Watch this video explaining the relationships between source code, byte code, and machine code

Video 3: Control Flow and Variables
Watch this video about control flow and variables in three different program representations

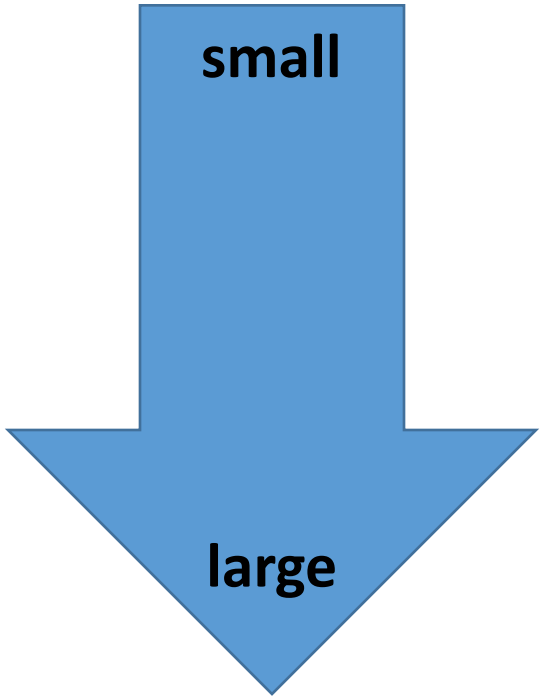
Video 4: Source Code, Byte Code, Operand Stack, Local Variable Table
Watch this video providing the fundamentals on how Java byte code gets executed

Video 5: Execution
Watch this video to see how Java byte code gets executed step by step

Video 6: Mapping back to Source
Watch this video to learn how to map from byte code to source code

Program Representations

- Instructions
- Basic blocks
- Control flow graph
- Functions
- Call graph
- OO: Classes
- OO: Class hierarchy graph
- Program



IA-32 Instructions

Native code instructions for Intel processors

IA-32 with GNU Assembler

- Intel Manuals
  http://www.intel.com/products/processor/manuals/
  - Volume 1: Basic Architecture
  - Volume 2: Instruction Set Reference
  - Volume 3: System Programming Guide
  - Volume 4: Model-Specific Registers
  - Optimization Reference Manual
- GNU Assembler (as) Manual
  http://sourceware.org/binutils/docs/as/index.html
  - Section 9.16: 80386 Dependent Features

What’s this?

movb %al, 7(%ebp)

target, source

Intel vs. AT&T IA-32 Assembly Syntax

Intel Syntax

- mnemonic source, target
- Register
  - eax
- Immediate
  - 7
- (Memory) Operand size
  - mov al, byte ptr 7(%ebp)

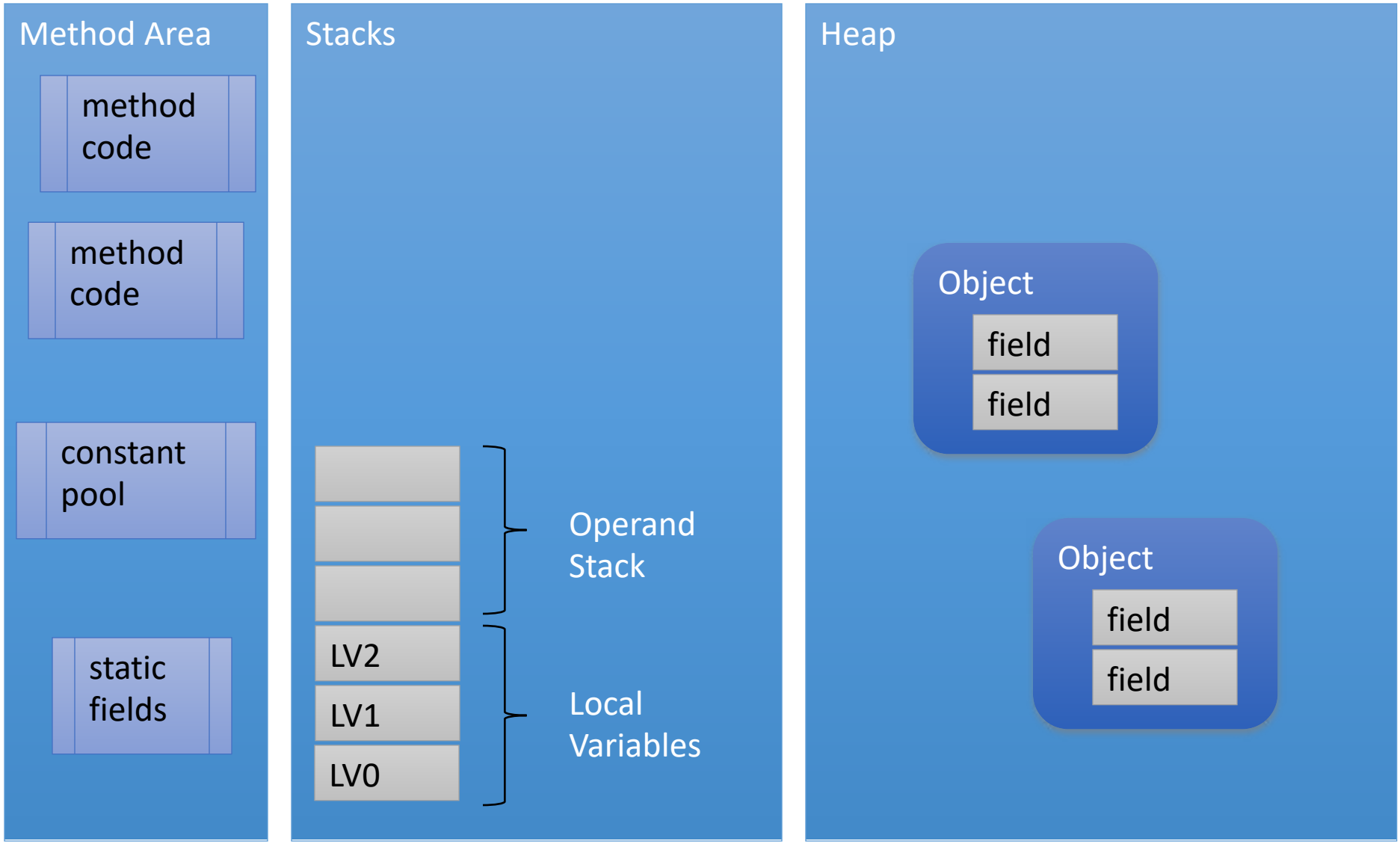
AT&T Syntax (GNU)

- mnemonic target, source
- Register
  - %eax
- Immediate
  - $7
- Operand size
  - movb %al, 7(%ebp)

JVM Memory

Where to store data

JVM Memory Organization



Types

How to constrain data

Types in the Java Language

Tell me…

Types in the Java VM

- Primitives
  - boolean (int – Z)
  - byte (int – B)
  - char (int – C)
  - short (int – S)
  - int (int – I)
  - float (float – F)
  - long (long – J)
  - double (double – D)
- References
  - Classes & interfaces – Ljava/lang/String;
  - Arrays – [I, [Z, [Ljava/lang/String;, [[I

Data, Types, & Operations in Different Layers

	Source Code	Byte Code	Machine Code
Locations	Local Variable Parameter Static Field Instance Field	Operand Stack Local Variable Table Static Fields Instance Fields	CPU Registers RAM Memory
Types	boolean byte char short int float long double java.lang.Object int[]	Z B C S I F J D L…; […	
Operations	**Operators:** +, &&,?:, ., [], …	**JVM Instructions:** iload, aload, dadd, ireturn, getfield, …	**Machine Instructions:** movb, movw, movl, movq …

JVM Instructions

How to operate on data

Java VM Instruction Set

Authoritative Source:

JVM Specification
Section 6.5. *Instructions*
http://docs.oracle.com/javase/specs/

Let's Play Compiler!

https://miro.com/app/board/o9J_lv8i1yQ=