

lecture 25

Java Virtual Machine

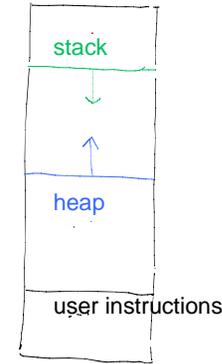
(different from MIPS virtual machine)

Wed. April 13, 2016

- what is the JVM ?
- what is Java "byte code"
(is it related to MIPS assembly/machine code ?)
- what are .class files ?
- what is garbage collection?

MIPS Virtual Machine

what is the Java analogy ?



invoked methods and local variables ?

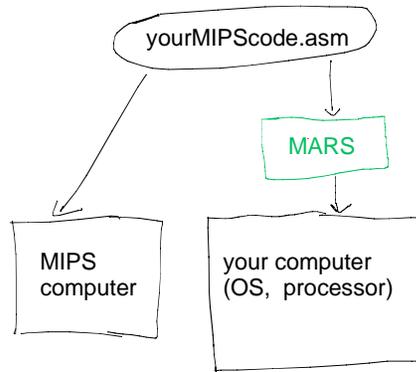
objects and garbage ?

class definitions ?

"Virtual Machine" (MIPS, Java) can refer to:

- 1) an abstract specification
 - a well defined instruction set (both assembly and machine code)
 - specified "on paper"
- 2) an implementation
 - software that runs on your computer (e.g. MARS for MIPS)
- 3) a runtime instance of that software

Example: what you've seen in COMP 273



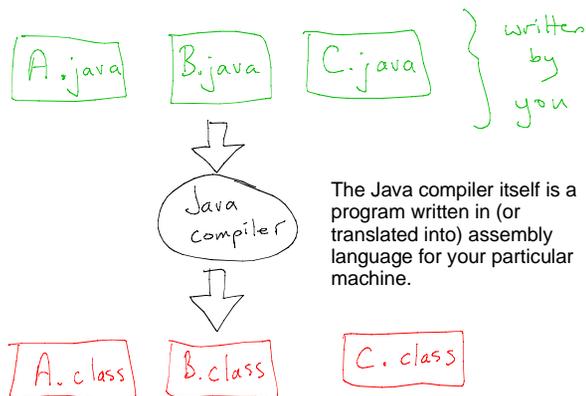
MARS is written in high level language (Java) and compiled to Java byte code.

It is a MIPS virtual machine running on a Java virtual machine.

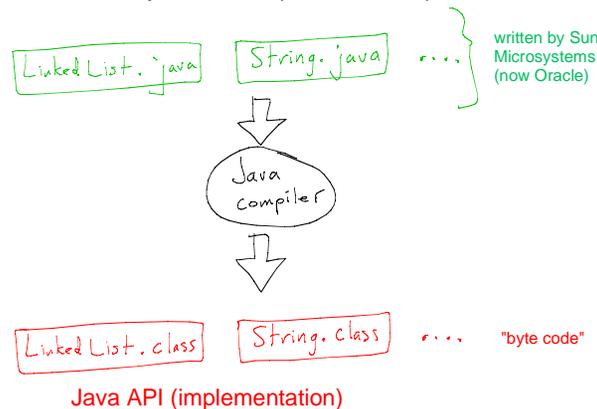
What does that mean?

Let's turn our attention specifically to the Java Virtual Machine (JVM).

Compile time (your programs)



Compile time ("Java API")



"Java API" can refer both to:

- 1) an abstract specification
 - set of classes and interfaces (with methods & fields)
 - on paper
- 2) an implementation (libraries)
 - i.e. class files (see previous slide)

Run time

your compiled classes

Java API classes



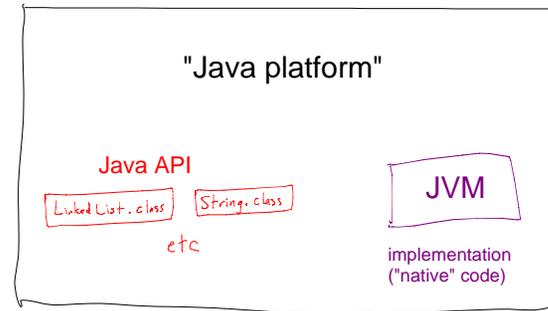
The above classes are "loaded" into the JVM at runtime, as they are needed.



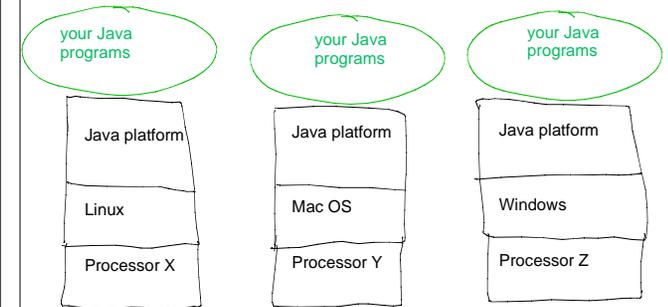
your compiled code



"Java platform"



Portability



The Java platform and the OS both can be written in a higher level language (C, C++) but ultimately they must be compiled down to assembly language or machine code for particular processor.

Java History (early 1990s)

- Java language created (by James Gosling at Sun)
- first web browsers written (html created) (*Java applets run in browser*)
- Mosaic (1993), Internet Explorer (1995)

Why was Java so important for www ?

- portability

(downloaded .class files can run on any JVM)

- security

(your compiled Java code doesn't know which computer it is running on. The platform provides a layer of protection (unlike C code which uses memory addresses explicitly)

Let's compare MIPS with Java (and then JVM)

MIPS (assembly)

Java (high level)

data

- registers
- Memory
 - stack (0xffff ffff)
 - globals & heap (0x1000 0000)

classes

- fields (data)
- methods (instructions)
- superclass

objects (instances of classes)

instructions

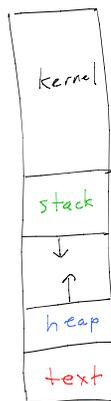
MIPS

data

- registers
- Memory
 - stack (0xffff ffff)
 - globals & heap (0x1000 0000)

instructions

- Memory
 - text

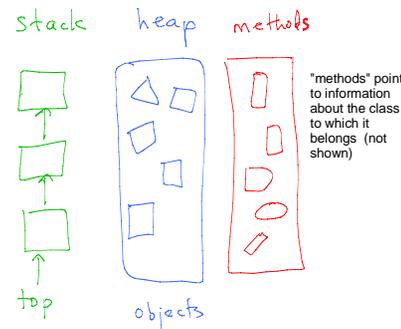


JVM

Everytime a method is invoked, a stack frame is added.

Stack frames correspond to invoked methods, like in MIPS. Stack frames contain:

- a pointer to a method
- a program counter within that method. (This is quite different from what you are used to in MIPS.)
- local variables of the method, e.g. references to objects.
- an operand stack (a stack within a stack) which I will explain later.



Details are very implementation dependent. Think of the above as data structures in a JVM, namely in assembler code of the real computer.

- what is the JVM ?

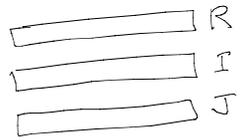
- what is Java "byte code" ?

(is it related to MIPS assembly/machine code ?)

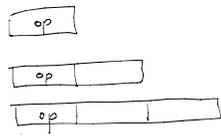
- what are .class files

- what is garbage collection?

MIPS instructions



JVM instructions ("byte code")



opcode (one byte)

http://en.wikipedia.org/wiki/Java_bytecode_instruction_listings

MIPS

registers and Memory

```
add  --,--,--
beq  --,--,--
jr   --
lw   --,--
sw   --,--
```

JVM

No registers (except PC)

Indeed, no CPU.

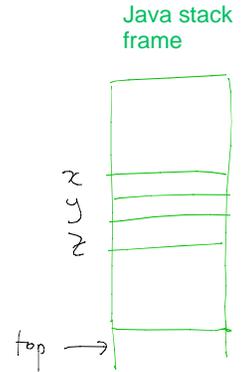
Uses "operand stack" (stack within a stack)

Example 1

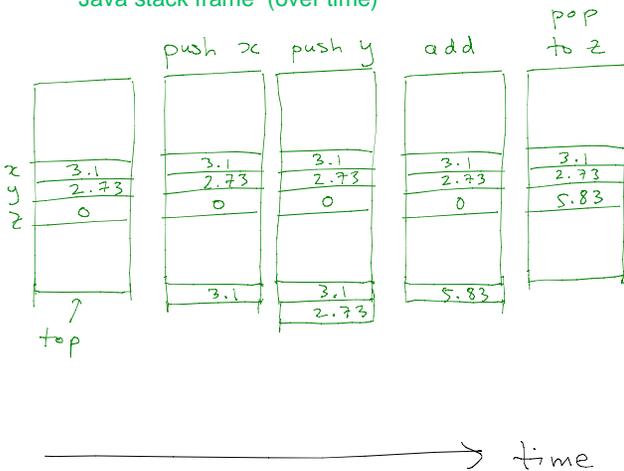
```
float x, y, z
:
z = x + y
```

How is this done?

MIPS analogy: load words from Memory into registers, then perform operation, then write back to Memory.



Java stack frame (over time)



local variable index in the method's stack frame

push x
push y
add
pop to z

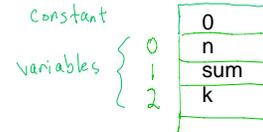
load_0
load_1
fadd
fstore_2

opcode

0x22
0x23
0xb2
0x45

Example 2

```
class SumToN {
    public static int sumton(int n){
        int sum = 0;
        for (int k=0; k < n; k++){
            sum = sum + k;
        }
        return sum;
    }
}
```



0: iconst_0
1: istore_1
2: iconst_0
3: istore_2
4: iload_2
5: iload_0
6: if_icmpge 19
9: iload_1
10: iload_2
11: iadd
12: istore_1
13: iinc 2, 1
16: goto 4
19: iload_1
20: ireturn

Try it yourself (linux)

javac SumToN.java

produces SumToN.class

javap -c SumToN

"disassembles" the class file

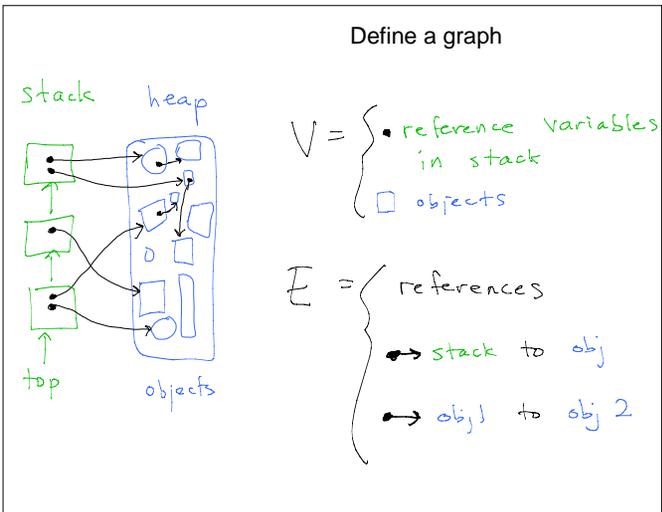
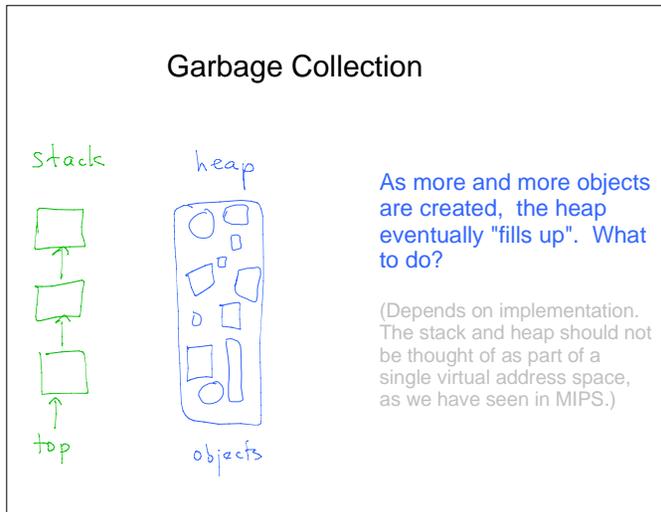
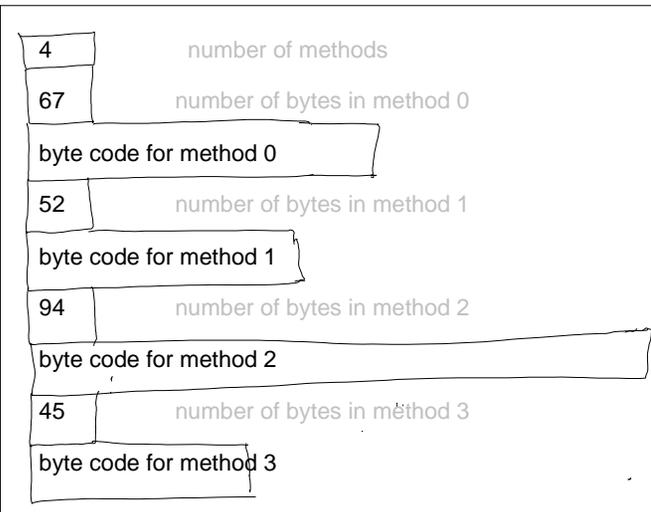
What defines a class ?

- name
- fields (identifier, type)
- methods (local variables, instructions, return type)
- modifiers

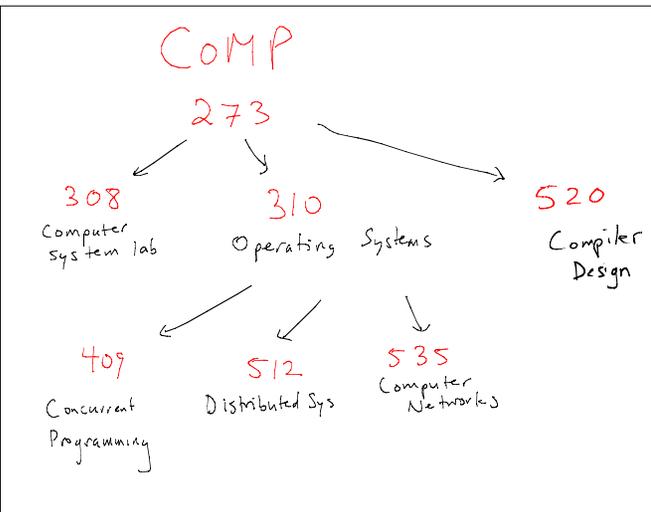
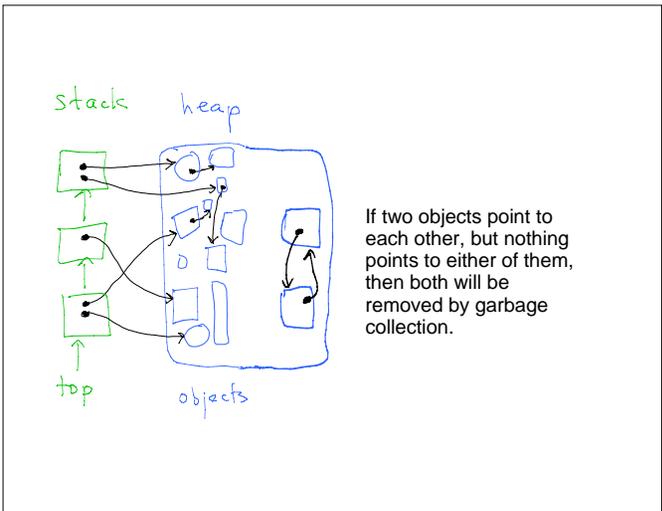
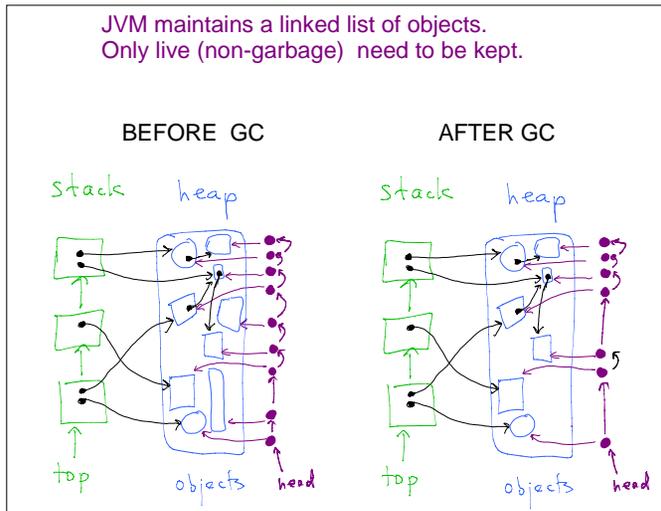
What is in a class file?

What is in a class file? (on disk)

- 4 bytes 0xCAFEBAFE
- constants e.g. numbers, strings, identifier (names)
- fields (including types)
- methods (byte code)



- ### Garbage Collection: "Mark and Sweep" Algorithm
- mark each **object vertex** as not visited
 - for each reference variable (vertex) in the **stack**
 - traverse the graph starting from that vertex, and mark each **object** that you visit as visited
 - remove each not visited object vertex (garbage)



- ### Final Exam
- 48 multiple choice questions (5 choices per question):
- 18 have number / bitstring / hex / boolean formula answers
 - 6 are of the form 'which of these is *not* correct?'
 - the rest require you to choose the best (positive) answer e.g. which of the following is correct?

Final Exam

48 multiple choice questions.

Answer them all. If you are unsure, then eliminate as many as you can and guess from the rest.

This approach is consistent with principle of "no negative marking": <http://www.ugent.be/en/education/degree/practical/studentadmin/OEREnglish/multiplechoice.htm>

The issue is subtle. Some multiple choice exams penalize you for getting wrong answer, but they don't penalize you for leaving a question blank. This encourages you to do probabilistic calculations on the exam, based on your certainty of whether you are correct or not. This is a distraction (not good).

Instead of penalizing you for guessing, I will scale the grades downward, so you need to answer correctly more than half the questions correctly in order to get a grade of 50%. To my knowledge, such scaling is *not* controversial.