

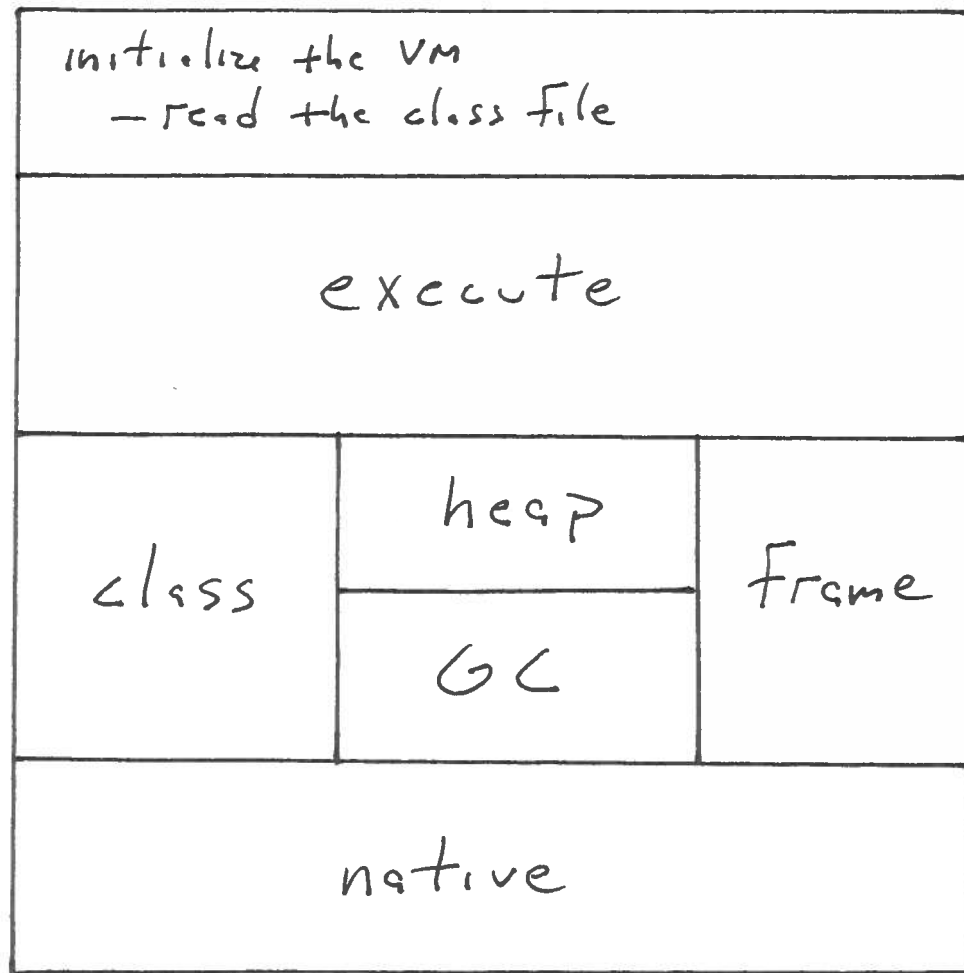
Reading mate Class Files

CS 520

Dept. of Computer Science

Univ. of New Hampshire

reading the class file is step #1 in implementing
the msc virtual machine



read the whole class file into memory

— place into an array of 32-bit integers
↳ the VM memory

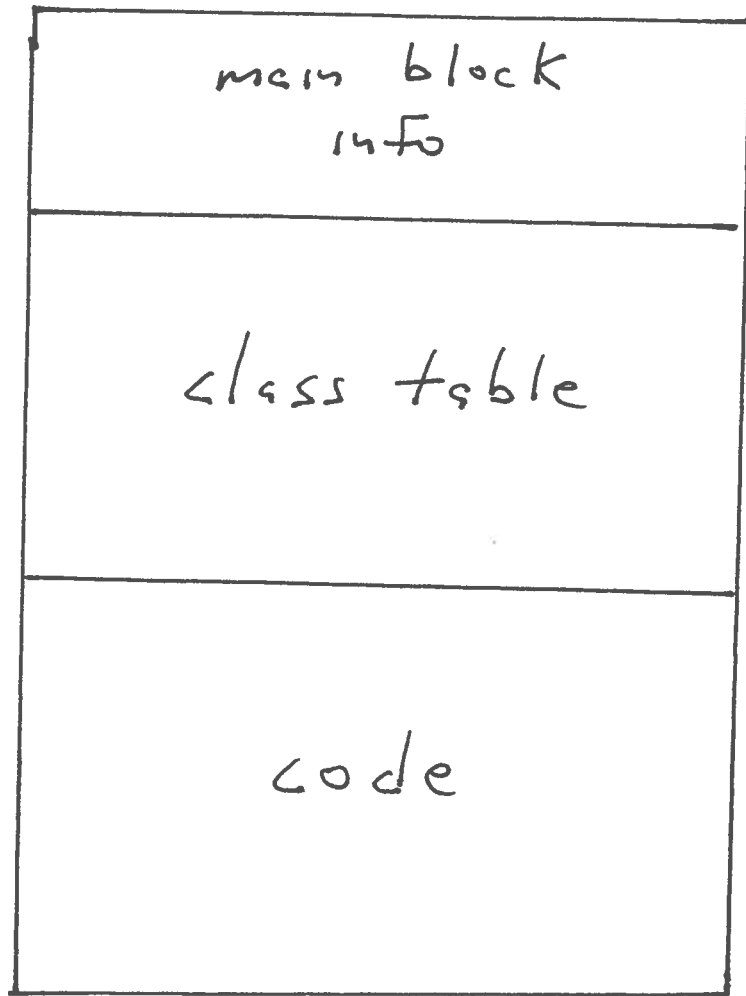
— i.e. deal with Endianness

— read the class file twice?

— once to get its length

— second time after allocating the
VM memory

implement functions to provide access to
the information in the class file



main block info

getMainBlockAddress: return VM-memory[0]

getMainBlockNumLocals: return VM-memory[1]

class table

classes \swarrow VM-memory [2]
(class descriptor address^{*}, class name^{**})
⋮
[class descriptor]
⋮

```
is Valid Class (class)
getObjectClass() #0
get Integer Class () #1
get String Class () #2
get Table Class () #3
get Ith Class (#int i)
is Object Class (class)
is Integer Class (class)
is String Class (class)
is Table Class (class)
```

* address is a byte-offset in the class file

** class name is a sequence of 32-bit words
— one ASCII character per word
— zero word to terminate

class descriptor

super class

fields

method table

getSuper(class)

getNumFields(class)

method table

#methods

(code address, # locals, method name*)

⋮

getNumMethods (class)
getMethodAddress (class, index)
getMethodNumLocals (class,
index)

* stored like a class name

code

getWordFromCode(address)

VM_memory [address / 4]

implementation suggestions

remember: addresses must be translated from
byte offsets to word offsets

you will need code to skip over the same strings

put in sanity checks:

is the class valid?

is the method index valid?

is the code address valid?