CS 201 Lecture 16
Program Stack and Calling Conventions
Spring 2014
Plan for today

- Memory spaces and their uses
  - Stack, heap, global
- Finishing up our sorting example
What do programs need memory for?

• Objects
• Temporary variables for computing functions
  – if you don’t have enough registers
  – Even if you do, what happens when call another function gets called?
    • Remember the return address problem

?
Different needs for different memory uses

• Objects are allocated somewhere, but float around until they’re not needed anymore
  – No way to know when we will be “done” with an object

• Temporary storage for functions has a well-defined usage timeline
  – Need it when the function starts, don’t need it anymore when the function returns

• It is more efficient to have different “sections” of memory for these different purposes
Stack and Heap

• “The Stack” or the program stack, is used for dynamically giving functions the space they need to execute and call other functions

• The Heap is the region of memory where objects are allocated
  – Any call to “new” grabs memory from the heap
Calling functions

What do we need to save when we call another function?

• Registers that we might need or don’t want other function to “clobber”
• Return address
• Function arguments

If different people are writing a function and a call to that function, they have to agree on how the function should be called
Back to Sorting Example
What do programs need memory for?

• Objects
• Temporary variables for computing functions
  – if you don’t have enough registers
  – Even if you do, what happens when call another function gets called?
    • Remember the return address problem
• The code
  – Code memory (yes, really)

?
Other memory spaces

What happens when you type

```java
System.out.println("Hello");
```

- “Hello” has to come from somewhere
- Could use instructions to create the string
- More efficient to put the string “Hello” somewhere in memory before the program starts
Static / Global Memory

- Data/objects that are not dynamically created
  - Static variables in Java or other languages
  - String literals

Recap:
- Stack, Heap, Code, Static
- System-level “data structures”
  - Identify categories of memory usage, and provision each with appropriate space and management policy