CS 161
Intro to CS I
Finish Recursion/
Begin Heap and Arrays
Practice Recursion...

- Write your own recursive `int pwr()` function that takes two integers as arguments and returns the integer result.
  - What does the function prototype look like?
  - Now, write the function definition...

```c
int pow(int base, int exp) {
  int result = 1;
  for (; exp > 0; exp--) {
    result *= base;
  }
  return result;
}
```
#include <iostream>

using namespace std;

// pre-condition is base and exp >= 0
int pow_r(int base, int exp)
{
    // base case
    if(exp==0)
        return 1;
    else
        return base * pow_r(base, --exp);
}

int main()
{
    cout << pow_r(2, 4) << endl;
    return 0;
}
Variables vs. Pointers

• Value Semantics
  – Values stored directly
  – Copy of value is passed
  int i, j=2;
  i=j;

• Pointer Semantics
  – Address to variable is stored
  – Copy of address is passed
  int *i, j=2;
  i=&j;

```
int i, j=2;
i=j;
```

```
int *i, j=2;
i=&j;
```
What if we don’t have the j?

• We need to create the address space.
• How do we do this?
  – new type;
• For example:
  int *i;
  i = new int;  //new returns an address
  *i = 10;
• http://cslibrary.stanford.edu/104/
Stack vs. Heap

- Static vs. Dynamic

compile time

runtime

memory

heap

stack

100
Static vs. Dynamic

- **Static Semantics**
  - Assign address of variable
  ```
  int *i, j=2;
  i=&j;
  ```

- **Dynamic Semantics**
  - Create memory
  - Assign memory to pointer
  ```
  int *i=NULL;
  i=new int;
  *i=2;
  ```
What About Memory Leaks?

• What happens here...

... 
int main () {
    int *i=NULL; //created in main function
    while(1) {
        i = new int;
    }
}

Fixing Memory Leaks...

• What happens here...

...