ACM Open House

Come learn about our active projects and interest groups in ACM...

Meet people, get involved and eat free pizza!

Thursday, January 24, 7pm

1404 Siebel Center

Promoting Undergraduate Research in Engineering (P.U.R.E.)

Spring 2013 Information Session

Wednesday, January 23rd, 6-7pm

1404 Siebel Center

pure.engr.illinois.edu

Announcements

Course policies:

http://cs.illinois.edu/class/cs225

For general assistance:

http://piazza.com/class#spring2013/cs225

HW0 available, due 1/23 before lecture

MP2 available, due 2/5, 11:59p. EC: 1/29, 11:59p.

Fun and games with pointers: (warm-up)

```
int * p, q;
```
What type is q?_______________

```
int *p;
int x;
p = &x;
*p = 6;
cout << x;
```
What is output?_______________

```
cout << p;
```
What is output?_______________

Write a statement whose output is the value of `x`, using variable `p`: ____________

Pointer variables and dynamic memory allocation:

```
int * p;
```

Stack memory

loc	name	type	value
a40	p	int *	

Heap memory

loc	name	type	value

Youtube: pointer binky c++

Fun and games with pointers:

```
int *p, *q;
p = new int;
q = p;
*q = 8;
cout << *p;                                 What is output?_______________
q = new int;
*q = 9;
p = NULL;                                   Do you like this?______________
delete q;
q = NULL;                                   Do you like this?______________
```

Memory leak:

Deleting a null pointer:

Dereferencing a null pointer:

Fun and games with pointers:

```
int * p, * q;
p = new int;
q = p;
delete p;
… // some random stuff
cout << *q;
```

Do you like this?______________

Fun and games with pointers:

```
int * p; int x;
p = x;
```

Do you like this?______________

What kind of error?

Compiler Runtime

```
int * p;  int x;
```

Variable `p` can be given a target (pointee) in two ways. Write an example of each.

Use the letters S and H in a meaningful way to tell where the pointee exists in memory.

```
int * p;
*p = 37;
p = NULL;
*p = 73;
```

Do you like this?______________

What kind of error?

Compiler Runtime

```
int * p, * q;
p = new int;
q = p;
delete p;
… // some random stuff
cout << *q;
```

Do you like this?______________

Which of the following snippets are buggy?

```
int *p, *q;
p = new int;
q = p;
*q = 8;
q = new int;
*q = 9;
p = NULL;
```

```
int *p, *q;
p = new int;
q = p;
*q = 8;
delete q;
*p = 12;
p = NULL;
```

```
int *p;
int x = 5;
p = &x;
delete x;
p = NULL;
```

```
int *p;
int x = 5;
*p = x;
```

Stack vs. Heap memory:

```
void fun() {
  string s = "hello!";
  cout << s << endl;
}
int main() {
  fun();
  return 0;
}
```

System allocates space for s and takes care of freeing it when s goes out of scope.

Data can be accessed directly, rather than via a pointer.

```
void fun() {
  string * s = new string;
  *s = "hello?";
  cout << *s << endl;
  delete s;
}
int main() {
  fun();
  return 0;
}
```

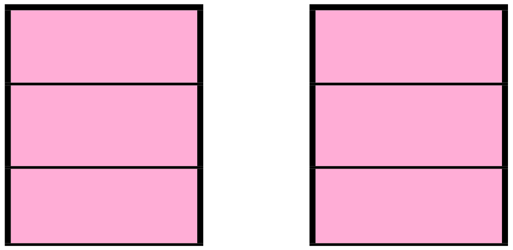
Allocated memory must be deleted programmatically.

Data must be accessed by a pointer.

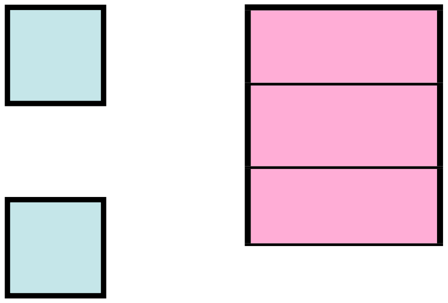
Pointers and objects:

```
class face {
public:
    void setName(string n);
    string getName();
    …
private:
    string name;
    PNG pic;
    boolean done;
};
```

```
face a, b;
… // init b
a = b;
a.setName(“ann”);
b.getName();
```



```
face * c, * d;
… // init *d
c = d;
c->setName(“carlos”);
(*d).getName();
```



Arrays: static (stackic)

```
int x[5];
```

Stack memory

loc	name	type	value

Arrays: dynamic (heap)

```
int * x;

int size = 3;
x = new int[size];

for(int i=0, i<size, i++)
    x[i] = i + 3;

delete [] x;
```

Stack memory

loc	name	value

Heap memory

loc	name	value

A point to ponder: How is my `garden` implemented?

```
class garden{
public:
…
// all the appropriate public members
…
private:
flower ** plot;
// other stuff
};
```