Announcements.

HW due Friday by start of class.

C8 180 - Lecture 4
Why outside of class?

Add Aplyal Function:

Add a friend such that

Count as my point. Example: makes sense to allow

instance of an object right operator to be

X, Y < deriv

ostream & operator << (ostream & out, Point p)
{
    return out<<p.gx()<<","<<p.gy()<<" \n";
}

Free-standing operator definition outside the formal Point class definition

ostream & operator >> (ostream & out, Point & p)
{
    in a few slices...
    return p.set((not in class)

Aplyal Function:
Child class inherits all attributes and methods

Ex: Rectangle vs Square

Parent class w/ child classes

Inheritance

What is it?
Example: Square class
Ex: public:

Inherited:

But: what about data which should not be private but also shown to be public is inherited and private is not.

A new type of data:

Other issues:

int Height;

protected:

int Weight;

int Weight


What does \( b = (3, 4) \) do?

In Python, variables are pointers to objects. Memory Management

\[ \begin{align*}
-1 &= \frac{y}{-5.8} \\
-x &= \frac{3}{2}
\end{align*} \]
C++: A more complicated setup...

C++ allows 3 different models for storing & passing information.

1. Value
2. Reference
3. Pointer

(Remember that strange & a few slides ago?)
This is more efficient both for space and speed.

$\begin{array}{c|c}
\text{y} & 7.0 \\
\hline
\text{x} & 5.0 \\
\hline
\text{b} : \text{Point} & \\
\end{array}$

$\begin{array}{c|c}
\text{y} & 0.0 \\
\hline
\text{x} & 0.0 \\
\hline
\text{c} : \text{Point} & \\
\end{array}$

Point $b$ (5, 7).

Point $c$ (5, 7).

Point $a$ (5, 7).

When a variable is created a preceeding value value variables.
If I change a, I'd also change b.

If they stay separate,

Different than Python:

<table>
<thead>
<tr>
<th>y</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Point C (11)'

\[ a = b + c \]

Now suppose we set \( a = b \).
So changes in function to pt don't affect myPoint!

Point pt(myPoint);

When someone calls isOrigin(Point myPoint)

{ return pt.x == 0 && pt.y == 0; }

bool isOrigin(Point pr)

functions: passing by value
will change the value of `c (and a),`

```
EX: c = b;
```
isOrigin( myPoint ).

In main:

Passing these functions, they are primarily used for:

Instead, they are primarily used for

In main program.

Reference variables aren't usually needed.

Passing by reference:
Passing by reference (cont.)

- Faster
- More memory efficient
- Any changes inside the function persist after the function

Why?
Compiler will ensure that it isn’t modified.

If next to input parameters -

```cpp
bool isOrigin(const Point3 pt) {
    return pt.x == 0 && pt.y == 0 && pt.z == 0;
}
```

Object modified, use const.

By reference, but don’t want access.

If we want the speed of processing...
3 return Vector Point; } Point operator+(const Point& other) const {
    return Point(operator+(x + other.x, y + other.y));
}

Another:

double distance(const Point other) const {
    return distance(other.x, other.y);
}

general: Speeding up the Point class:

Note: Can't make types Pointer Point.
in main already

Return type: ostream

Adding data: Change the stream by
we are changing the stream by
Note that we don't use cout since
Cannot be copied.
Here, G is required because streams

```cpp
return out;
```