Look for HTML on website soon.

- HTML due Friday

Announcements

CSE 216o - Classes in C++
Get line (can person)?

Context: What is your name?

Then: Shrink person

The next thing up to (but not including) the
get line is a function which stores

---

Get line
Problem:

1. Type: "15"
2. Input steam:
   - How old are you?
   - What would you like to eat?
3. Output:
   - 15 hot dogs
   - How hot are dogs?
mydata = open(fileName, 'r');
cin >> fileName; // cin >> fileName;
cout << "What is the file name?";
string fileName;
string mydata;
if (!not :)
    ifstream mydata(fileName, scores.txt);

if file is known:
    using namespace std;
    #include <fstream>
    #include <iostream>
    using File Structures - fstream
To append:

```
ofstream datastream("scores.txt", ios::app);
```

By default, writing to a file overwrites

(Think "w" in Python.)
Much more complex.

There is also an example of a single file which allows reading and writing to

Reading and writing
Displayed Age = "42.11".

age = 42

```cpp
int age = 42;
string displayage;
string stream;
stream << displayage;
stream << age;
stream << " is a string.";
Ex: Casting between numbers & strings.
```
3 // a is destroyed

count => "a" is "a" => "b" => "c" endl;

if b is destroyed
int b = 16;

else if b == 12;
int (a < 0)?
if a;
int mean(3);

int mean (2) ?
A note on variable scopes: Valid at A where is it
Valid at A
Unused -
```java
// Counter is destroyed
3 // x is destroyed

counter = 0
def x = value
end

for (float x = value; x > 2.5; x = x / 2)
    int counter = 0
    for loops:
    // Correct way:
```
Arrays as inputs to functions

Example: Write a function to specify if sum of values in an array is even.

```cpp
bool evenSum(int anArray[][], int size) {
    int sum = 0;
    for (int i = 0; i < size; i++)
        sum += anArray[i][i];

    return ((sum % 2) == 0);
}
```
Count as "the sum is even, "

To call: if (evenSum (myArray, length))

it like on carly.

prefer that it does - just use

doesn't carry whole carry but can

(while on those later)

\[ \text{Note:} \]

\[ \text{Note:} \]
Predefined operations/methods

Containers for data

What is a class?

Classes
Why
Never: `string("Hello")` greening

Why: Declaring a really dumb function
Never:
`string s();`

Optional input & initialization:
`string greening("Hello");`

Example:
Calling a constructor:
Creating an instance of a class
```cpp
void setY(double val) {
    y = val;
}

void setX(double val) {
    x = val;
}

double get() const {
    return y;
}

double getX() const {
    return x;
}

class Point {
public:
    Point() : x(0), y(0) {
        // Constructor
    }

private:
    double x;
    double y;
};

// Example:

void main() {
    Point p;
    p.setX(3);
    p.setY(4);
    // No explicit declaration of data members
}
```
Why is `shape` word only be in function?

This is done inside the class, but not inside a function. The class, but
decided, not just used in constructor.

- Data - public or private - is difficult.

- Classes:
2 Constructor Function

- name: same as object
  (only capital for you'll ever write)

- no return type — only fine

- can initialize variables via a list

  Point() : x(0), y(0) { }

  Point(double initialX=0.0, double initialY=0.0) : x(initialX), y(initialY) { }
Other differences

3) No self! Can just use -x or -y, it immediately scopes to the class attributes.

(There is a "this", but its usage is a bit more complex.)

4) Access control - public versus private. enforced by compiler.


```cpp
void setX(double val)
{
    x = val;
}

double getX() const
{
    return x;
}

Accessor Versus Mutator
```
start of Point class (semicolon is required)

{ }

{ return x * other.x + y * other.y; }
}

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

Point operator+(Point other) const
{
    return Point(x + other.x, y + other.y);
}

Point operator-(Point other) const
{
    return Point(x - other.x, y - other.y);
}

double operator/(double factor) const
{
    return scale(1/factor);
}

if (imag < 0)
{
    double mag = distance(Point(0, 0));
    void normalize();
}

void getXY()
{ return; } getXY()

else: otherwise:

only inside classes: add functional
In Python: `isinstance()(x)

No way to return 2 different types of versions of x: 
- default: \( x + y \)
- Point \( x = x + y \)

Other using operator: \( \text{in main} \)

1) \( x + \text{other} \rightarrow \text{allowed only inside the class} \)

Important things
a Point class should have to be

Why?

```cpp
// Display using form (x,y)
< new Point(x, y) out << "(" << x << ", " << y << ")";
// Invoke existing form with Point as left operand
Point p, q;
return q + p;
// Free-standing operator definition, outside the formal Point class definition
Point operator+(double factor, Point p) {
    ...;
}
```
Mother

Other

Functions

```cpp
public:

/*
Sometimes also have
pointers to
private members

class Point 3

private:

Separate class file: Point.h */
```