Objects IV

- Suppose we want to write a function --- outside of our dog class --- that given two dogs, determines which one is older.

  _________ older (_________, __________)

  What should the return type and the argument types be?

```
class dog {
	void setAge(double newAge);
	double getAge();
};

dog older(const dog & d1, const dog & d2)
{
	if (d1.getAge() > d2.getAge())
		return d1;
	else
		return d2;
}
```

None of the code in `older()` changes our dogs d1 and d2, which is good because they are marked const (so the compiler has to make sure they are not modified).

```
class dog {
	void setAge(double newAge);
	double getAge();
};

dog older(const dog & d1, const dog & d2)
{
	d1.setAge(100);
}
```

Now older() modifies a dog -- - this breaks the const label.

How can C++ tell what is OK to do with a const variable?

```
class dog {
      void setAge(double newAge);
      double getAge() const;
};
```

- Methods in a class that do not modify any of the class's fields should be marked **`const`**.
  - Otherwise these methods cannot be called on const variables (usually pass-by-const-reference arguments).

- What else should be marked **`const`** in the dog class?

LAB TIME!