Variable as a Remote Control

A useful memory aid

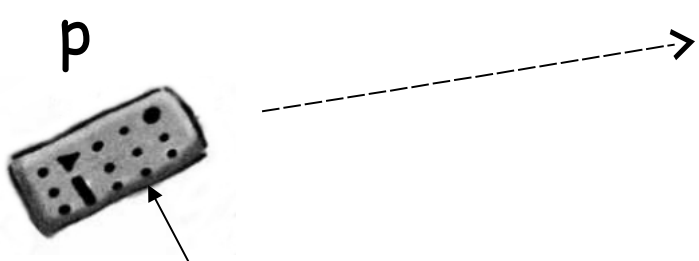
used in *Head First Java*

A Variable is a Reference

```
Person p = new Person( )
```

a *reference* for sending commands to an object

object



buttons on remote control are methods

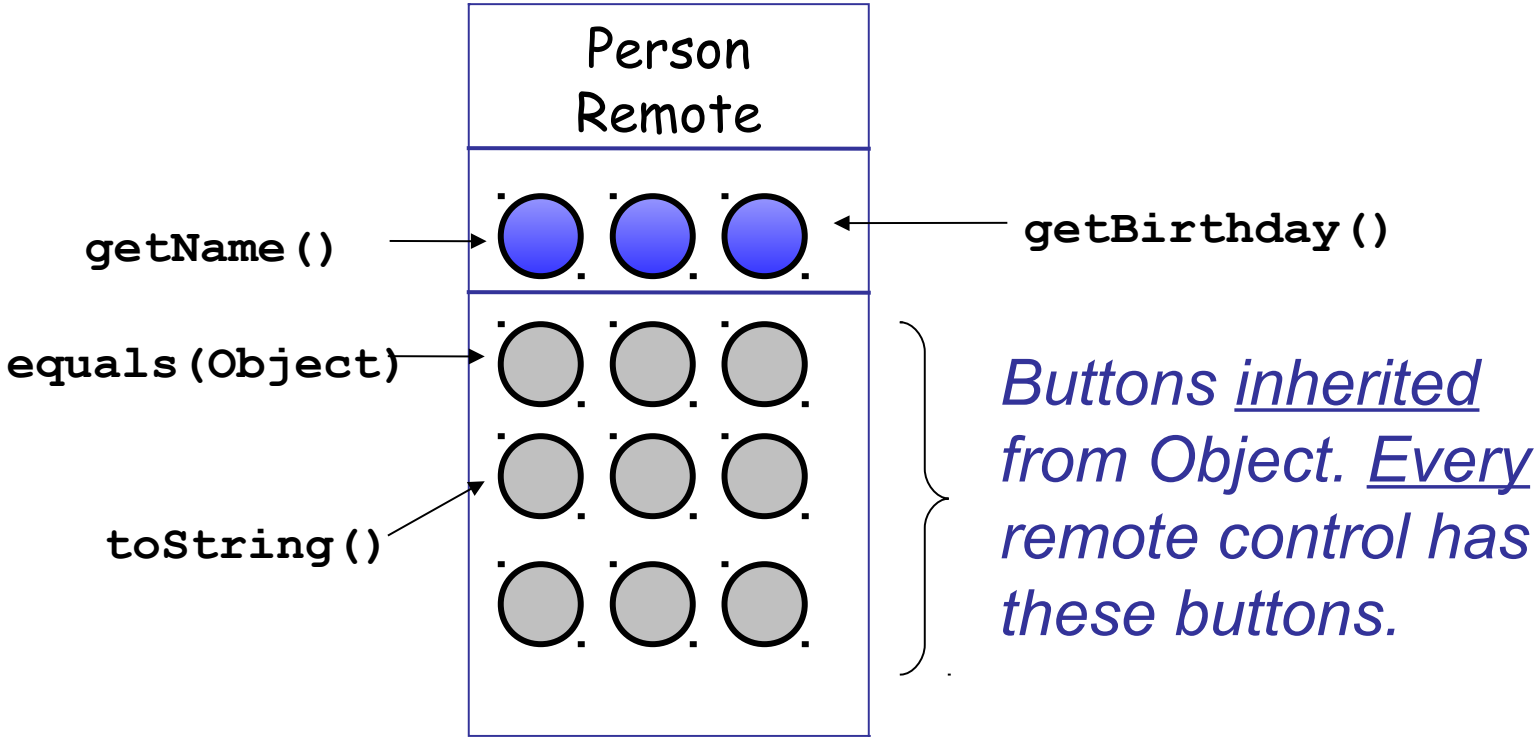
Person

```
#clone()
equals(Object)
finalize()
getClass()
hashCode()
toString()
getName(): Str
getBirthday()
```

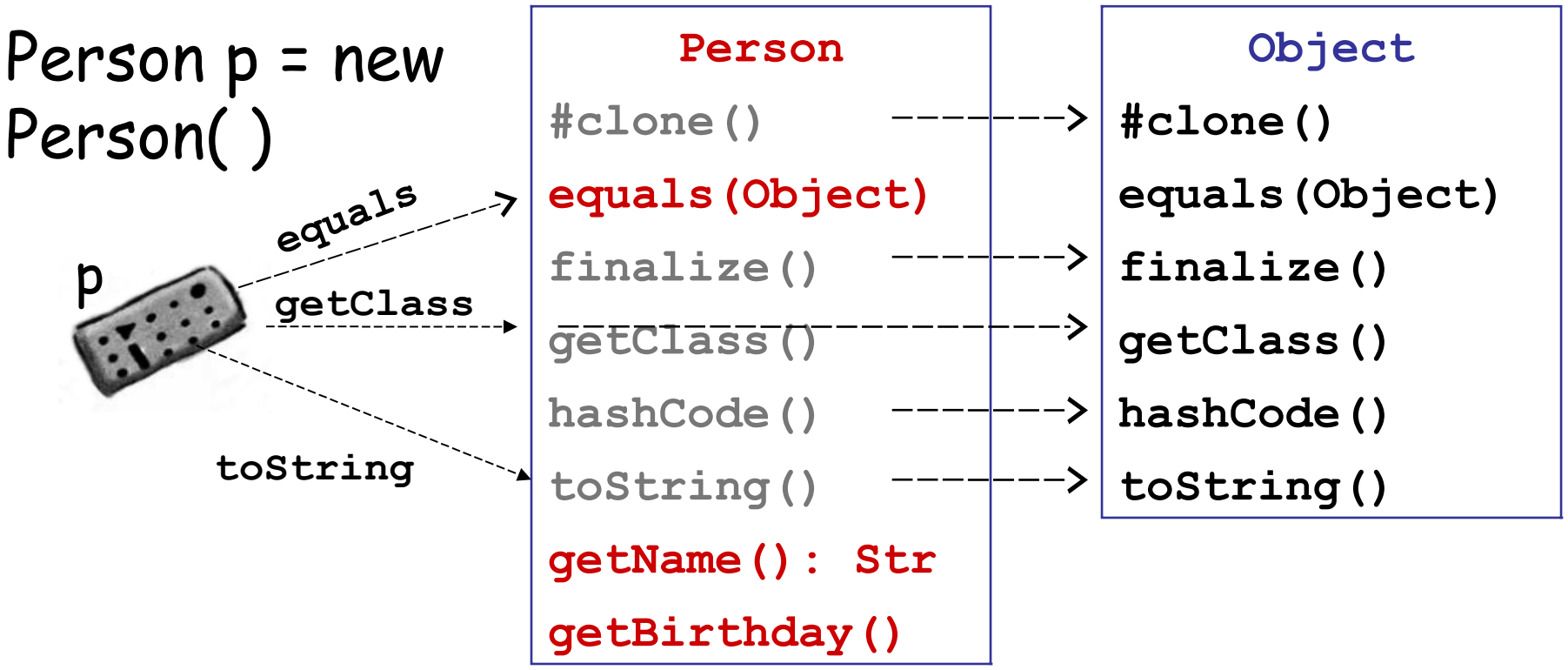
The Compiler decides what Buttons

Person p = _____

Compiler uses the *declared type* (Person) of variable to decide what buttons (methods) it has.



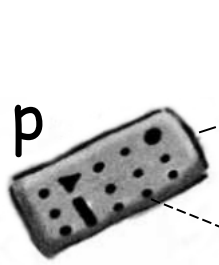
Invoking Methods



At runtime, JVM invokes method of actual object.
If a class *overrides* a method, the override is used.

Invoking Methods

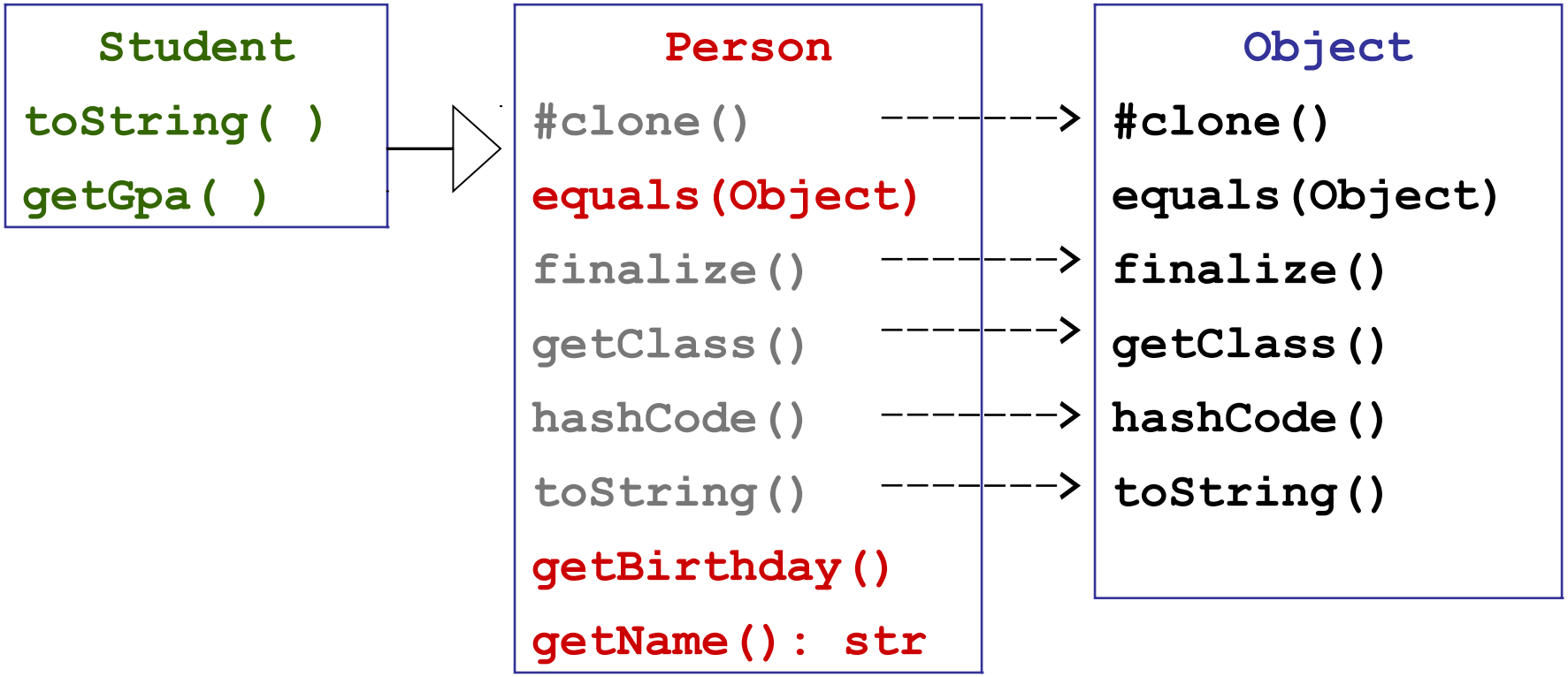
Person p = new Person( )



Person	Object
#clone()	#clone()
equals(Object)	equals(Object)
finalize()	finalize()
getClass()	getClass()
hashCode()	hashCode()
toString()	toString()
getName(): Str	
getBirthday()	

Person has equals(Object), to p.equals(...) uses that method. Person does not override getClass(), so p.getClass() invokes the method from Object.

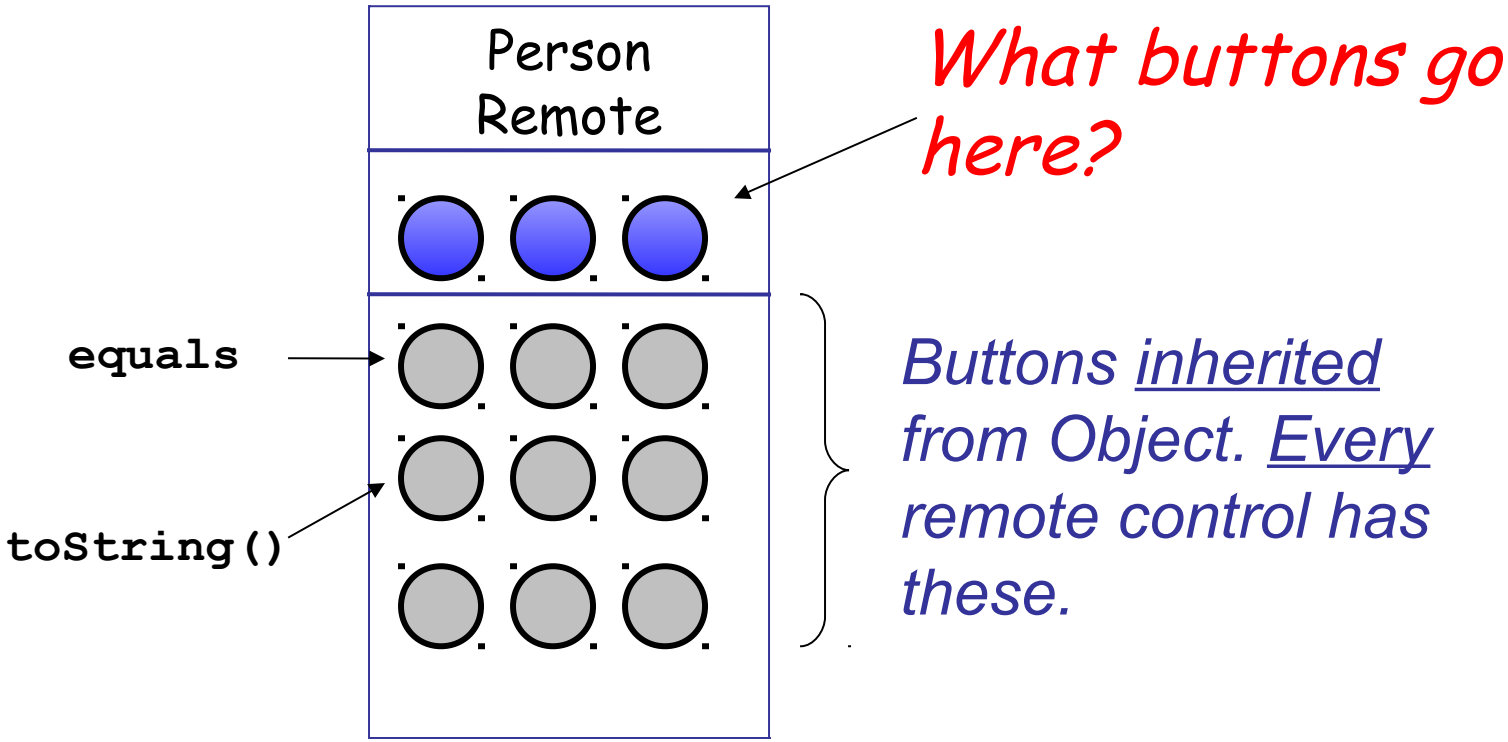
Student extends Person




```
class Student extends Person {
    public double getGpa() { . . . }
    public String toString( ) { . . . }
```

What Buttons Does `p` Have?

```
Person p = new Student( );
```

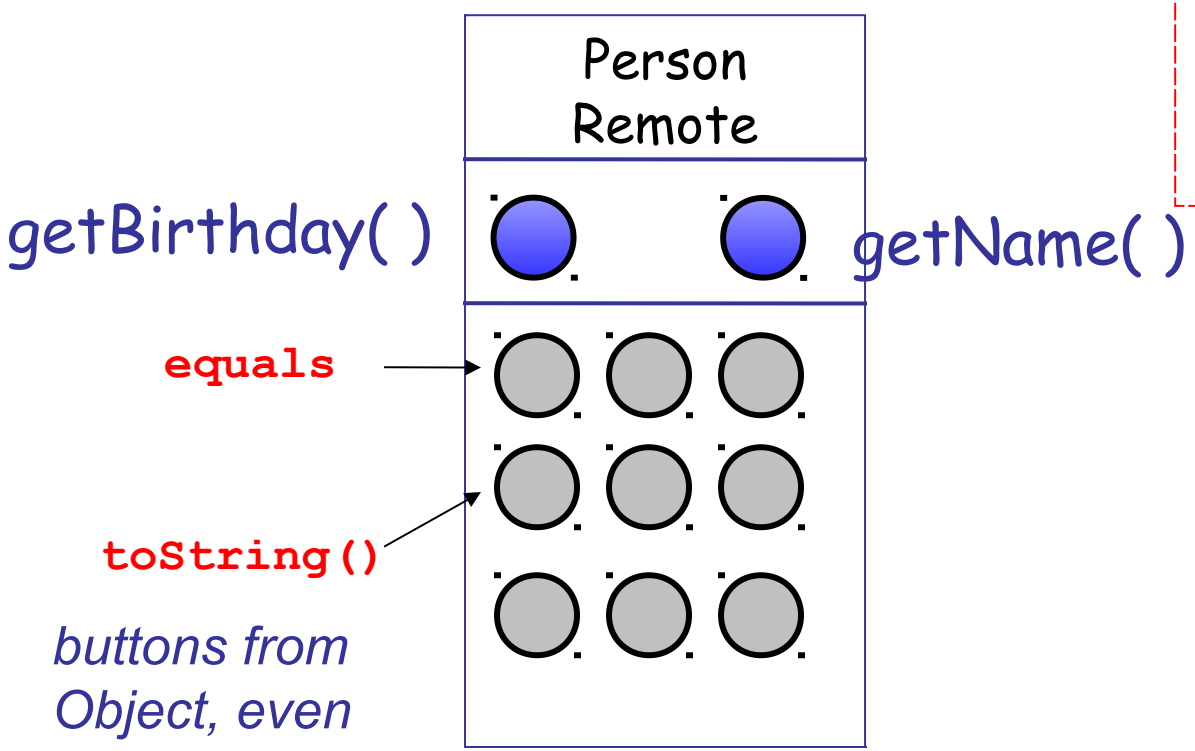


Does p Have a getGpa() button?

```
Person p = new Student( );
```

Student has a getGpa method.

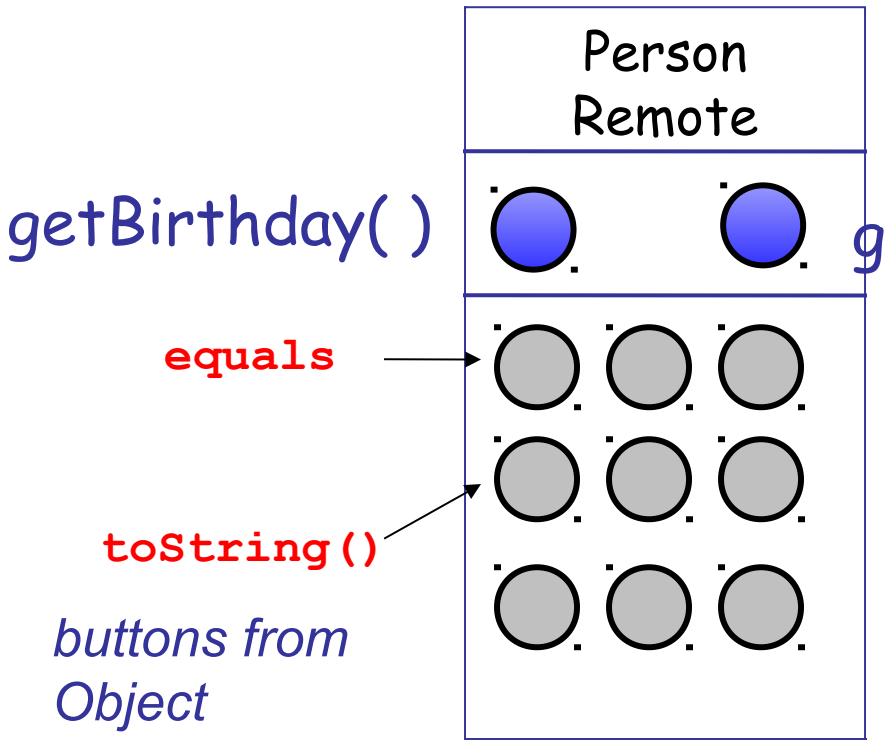
Why is there no getGpa button?



buttons from Object, even though definition is changed.

Does p Have a getGpa() button?

```
Person p = new Student( );
```



p does not have a getGpa() button because p is declared as a Person remote.

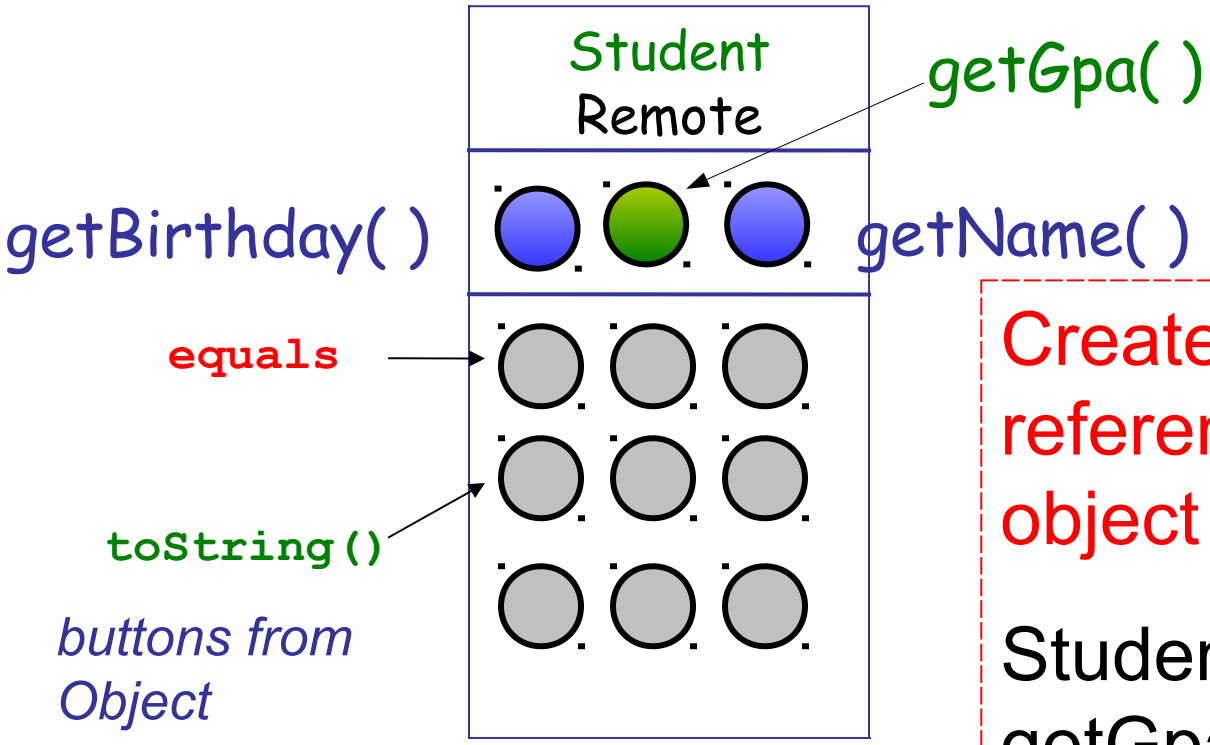
Person does not have a getGpa method.

How Can We Get a getGPA Button?

```
Person p = new Student( );
```

Change the Remote Control!

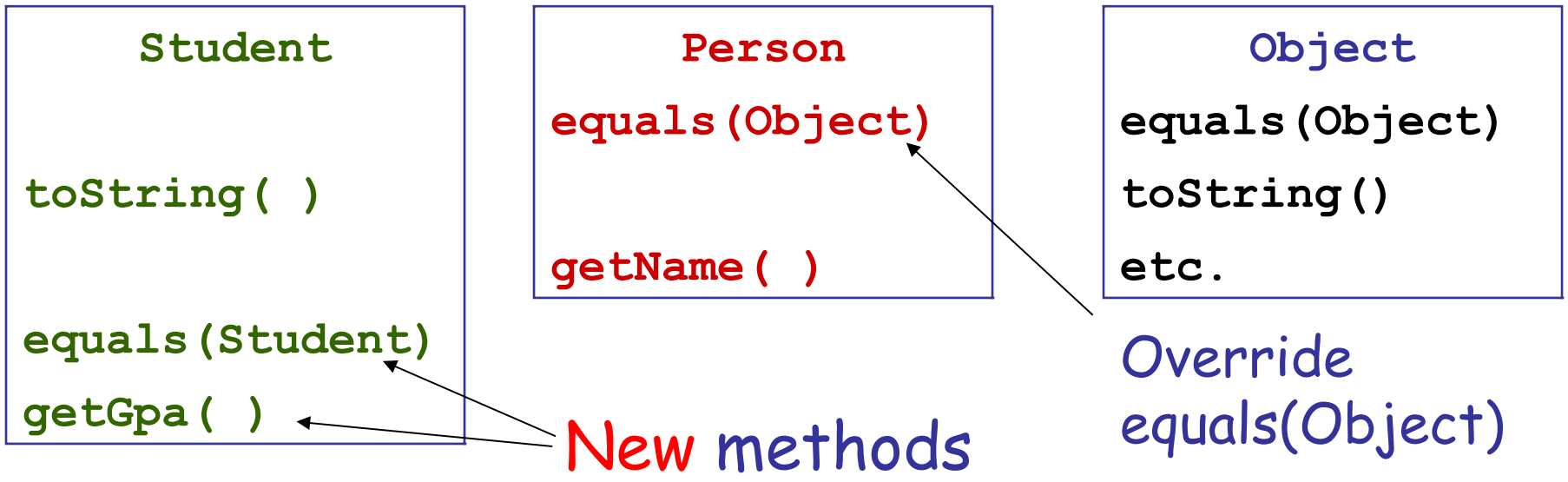
```
Person p = new Student( );
Student s = (Student) p;
```



Create a new reference to the same object using a *cast*.

Student s does have a getGpa method.

Method Signature includes Parameter




```
class Student extends Person {
    public boolean equals( Student s ) // BAD IDEA
    public String toString( )
```

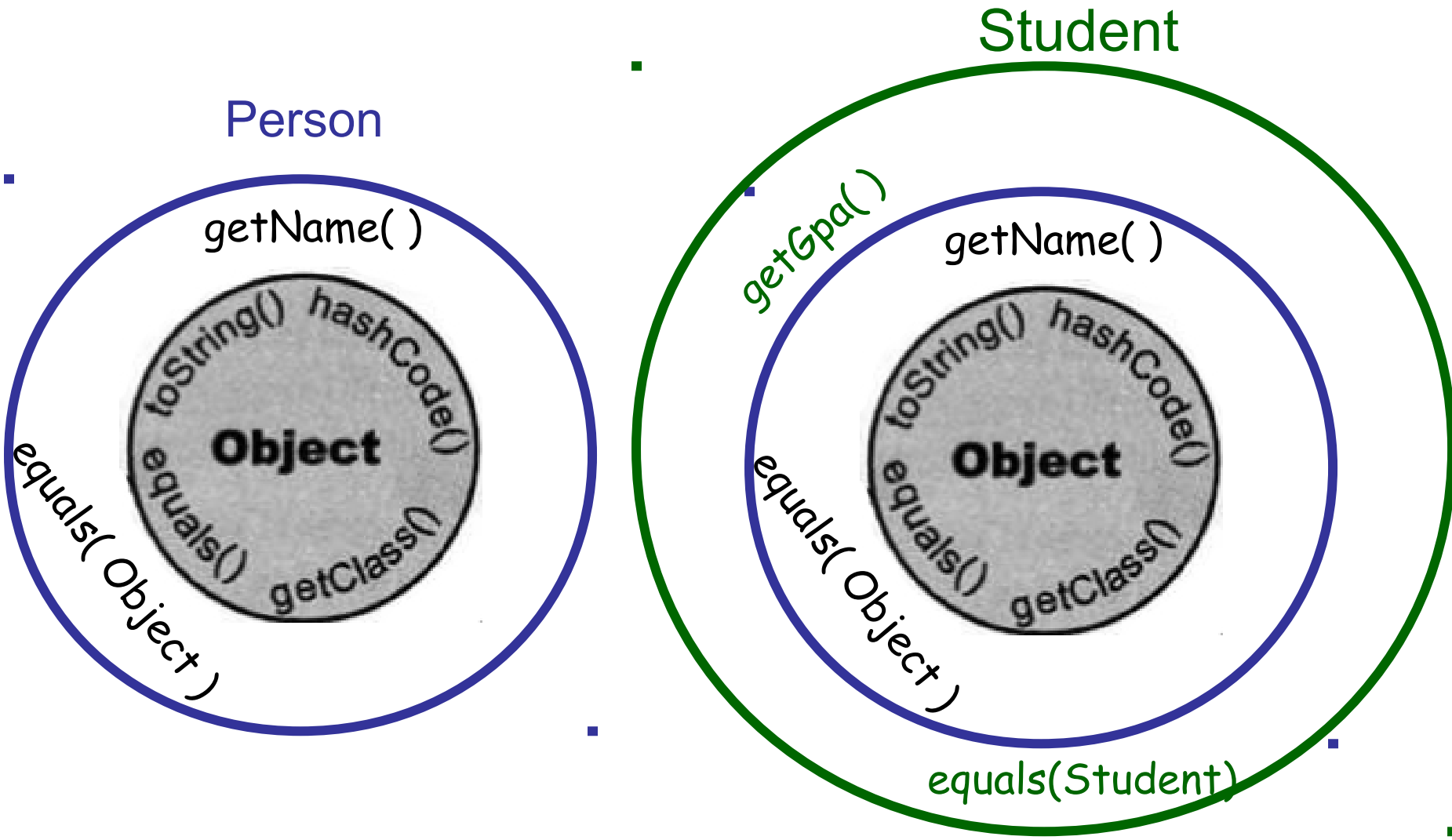
Which equals( ) is called?

Student	Person	Object
`toString( )`	`equals(Object)`	`equals(Object)`
`equals(Student)`	`getValue( )`	`toString()`
		`etc.`

```
Student a = new Student();
Person b = new Student( );
//1.
b.equals( a )
//2.
a.equals( b )
```

Draw the remote control !

Another view of Inheritance



Object References

Object obj = new Student();

obj.toString( ) ???

An "Object" remote control (reference) only knows the methods for object.