Car c = new Car()
c.equals(. . .)

CASTING OBJECTS

- We often write `equals` methods for classes
- Typically `public boolean equals(Object obj)`
- method can take in any object, not just type invoking it
- need to cast to the correct type

```
public boolean equals (Object obj) {
    if (obj instanceof Car) {
        Car c= (Car) obj;
        :
    }
```

CASTING OBJECTS

- How do we know it is of correct type?
  - first need to check
  - `obj instanceof Car` - will be true/false
- Once we know it is of correct type, cast it
  - `Car c = (Car) obj;`

```
double p = 10.5;
int x = (int) p;
```

INPUT/OUTPUT

- 3 standard ("system") I/O "streams"
  - standard input: `System.in`
  - standard output: `System.out`
  - standard error: `System.err`
- stream: think like a river or other flowing water

OUTPUT OPTIONS

- `print(...)`: options for most different types
- `println(...)`: prints whole line

Scanner scan = new Scanner(System.in)

INPUT OPTIONS

- `Scanner`: a parser
  - multiple constructors that allow inputStream, String, File
  - `next()` - tokenized by default whitespace
  - `nextLine()` - grabs whole line
  - `nextInt()`,`nextDouble()`
  - `hasNext()` - see if there is another token
  - `hasNextLine()` - is there another line
  - ways to use other delimiters

Scanner scan = new Scanner(System.in);

FILE INPUT

```
Scanner fileReader = new Scanner(new File(filename));
```

- works just like `Scanner` from `System.in`
- should put in try/catch in case file doesn't exist

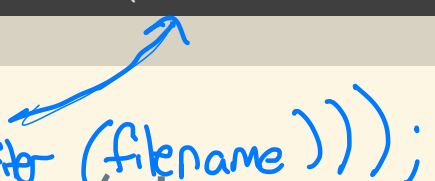
throwing
- throw new IllegalArgumentException();
- Can also be thrown automatically by java such as IndexOutOfBoundsExceptions

```
try {
    //code that might cause exception goes here
}
catch (Exception e) {

}
```

FILE OUTPUT

```
PrintWriter out = new PrintWriter(new BufferedWriter(new FileWrit
```



- nest in try/catch
- has all same methods as print stream (aka, `System.out`)