## Generic function programming

R's "S3" class system relies on *generic* functions. A call to a generic function is dispatched (by `UseMethod()`) to the appropriate class-specific method.

`class(x)` gets or sets the `charater` class name of object `x`. e.g. `class(3.14)`, `class(m <- lm(mpg ~ wt, mtcars))`

An S3 *method* has the name `GENERIC_FUNCTION_NAME.CLASS_NAME()`. e.g. `print()` is a generic and `print.lm()` and `print.data.frame()` are methods. e.g.

```
print # note call to UseMethod("print")
g = list(name="Margaret", age=2)
class(g) = "girl"
b = list(name="Philip", age=11)
class(b) = "boy"
print.girl = function(x) {
  cat(sep="", toupper(x$name), ", ", x$age, "\n") # girls get upper case
}
print.boy = function(x) {
  cat(sep="", tolower(x$name), ", ", x$age, "\n") # boys get lower case
}
print(g)
print(b)
```

e.g. `?abline` has a parameter `reg` that is "an object with a `coef` method."

```
plot(x=0:3, y=3:0)
coef.girl = function(object, ...) {
  return(c(object$age, 0)) # horizontal line with y-intercept age
}
abline(g)
```

In homework 2, we'll write a package that implements a class for which we'll include `print`, `coef`, and `predict` methods.

`methods(generic.function, class)` lists available methods for `generic.function` or `class`. e.g. `methods(print)`, `methods(coef)`, `methods(class="girl")`, `methods(class="lm")`

For more, see `http://adv-r.had.co.nz/OO-essentials.html` and the section "7 Generic functions and methods" in `http://cran.r-project.org/doc/manuals/r-release/R-exts.html`.