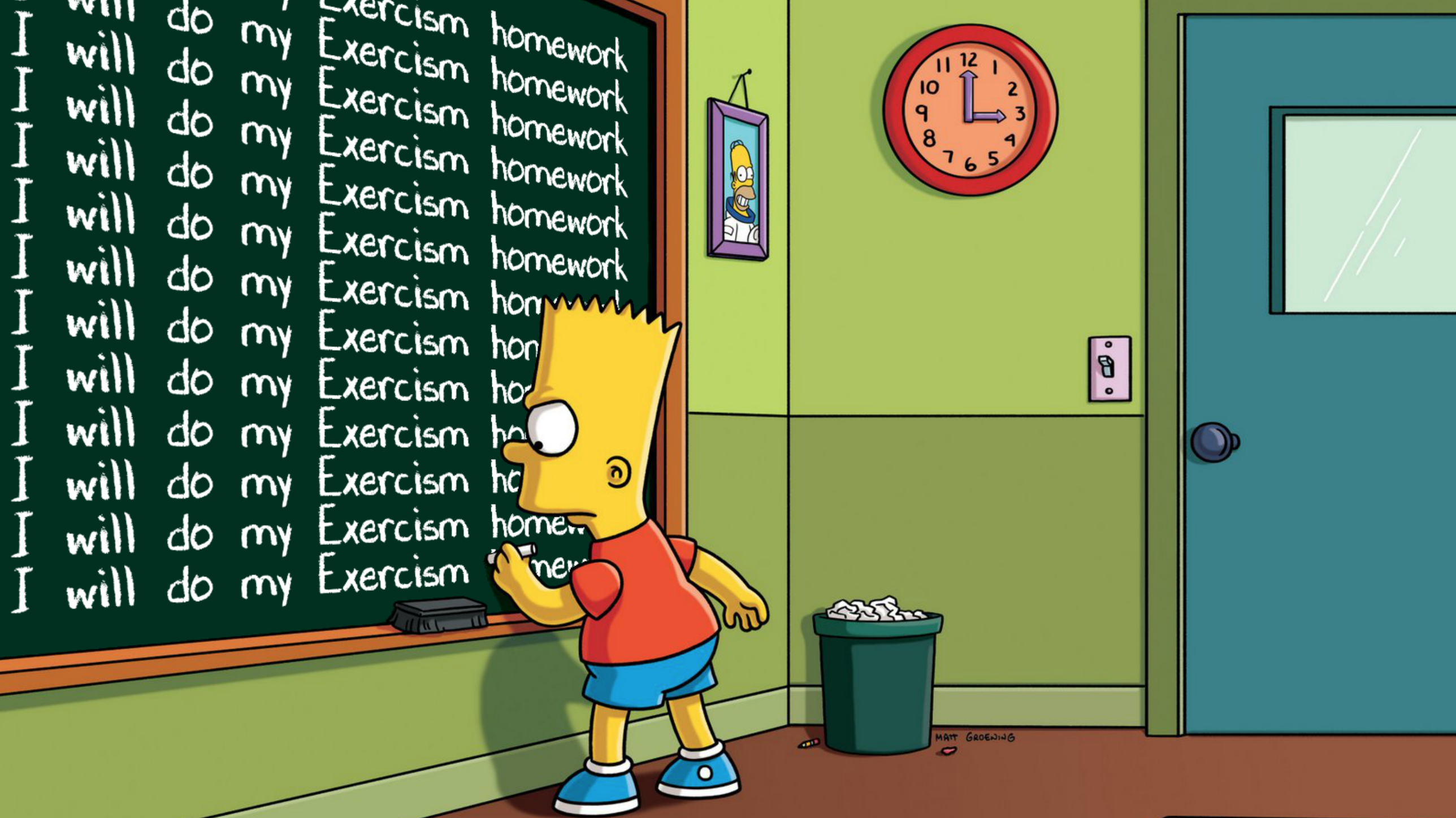
Exercism: Grade School


I will do my Exercism homework
MATT GROENING

School Roster

School Roster

School Roster

- 🎒 ***Add a student's name*** to the roster for a grade

School Roster

- 🎒 ***Add a student's name*** to the roster for a grade
- 🎒 ***Get a list of all students*** enrolled in a grade

School Roster

- ***Add a student's name*** to the roster for a grade
- ***Get a list of all students*** enrolled in a grade
- Get a ***sorted list*** of all students in ***all grades***

Take 1

Create a School Roster

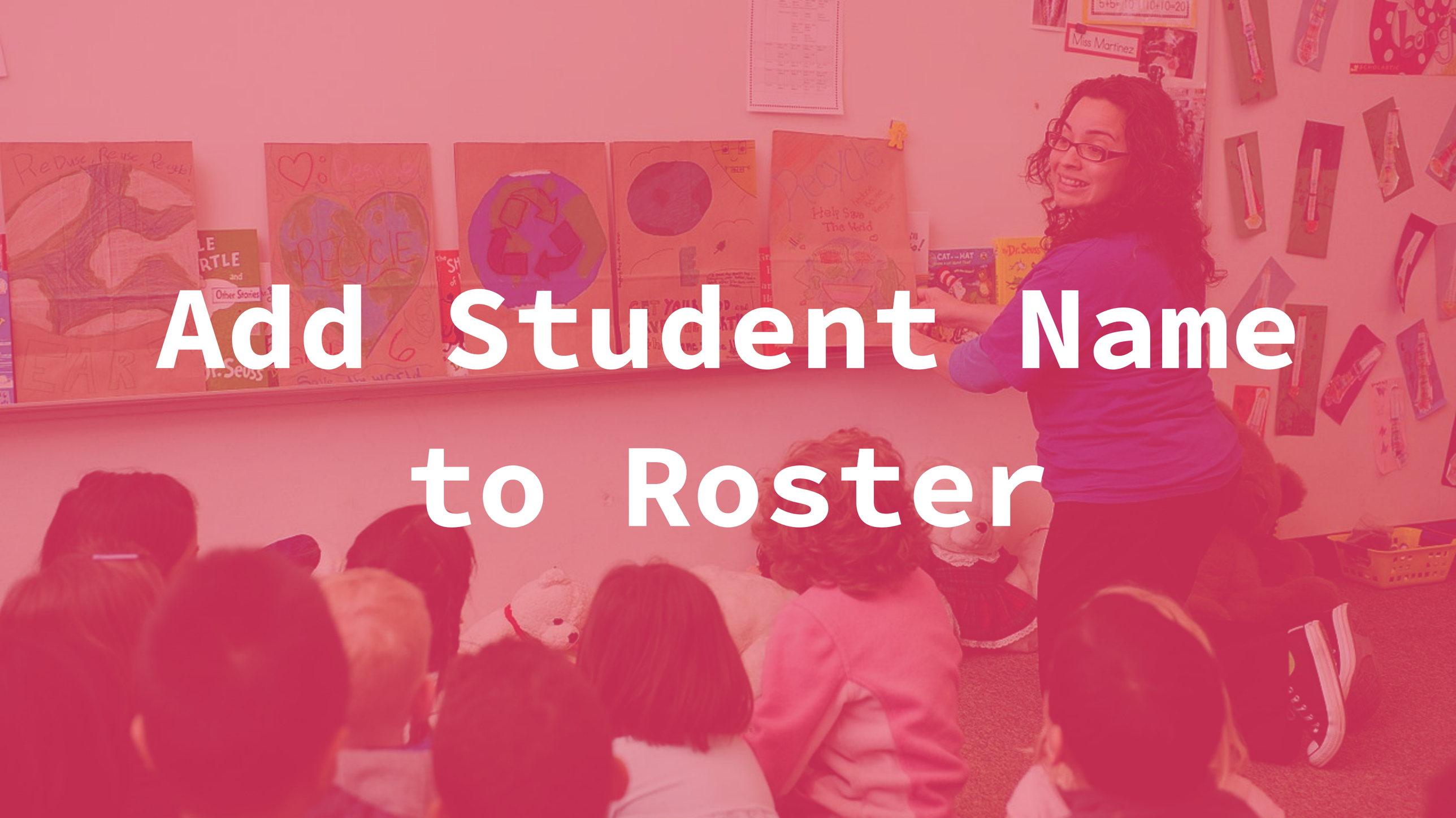
```
class School
  def initialize
    @roster = []
  end

  private

  attr_reader :roster
end
```

A Roster is filled with Enrollments

```
> @roster
=> [
      { grade: 1, students: ["Paul", "RobC"] },
      { grade: 2, students: ["Nick"]) },
      { grade: 3, students: ["Cath", "RobH"]) }
   ]
```



Add Student Name to Roster

Add Student Name to Roster

```
def add(name, grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students] << name
  else
    roster << { grade: grade, students: [name] }
  end
end
```

Add Student Name to Roster

```
def add(name, grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students] << name
  else
    roster << { grade: grade, students: [name] }
  end
end
```

Add Student Name to Roster

```
def add(name, grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students] << name
  else
    roster << { grade: grade, students: [name] }
  end
end
```

Add Student Name to Roster

```
def add(name, grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students] << name
  else
    roster << { grade: grade, students: [name] }
  end
end
```

Add Student Name to Roster

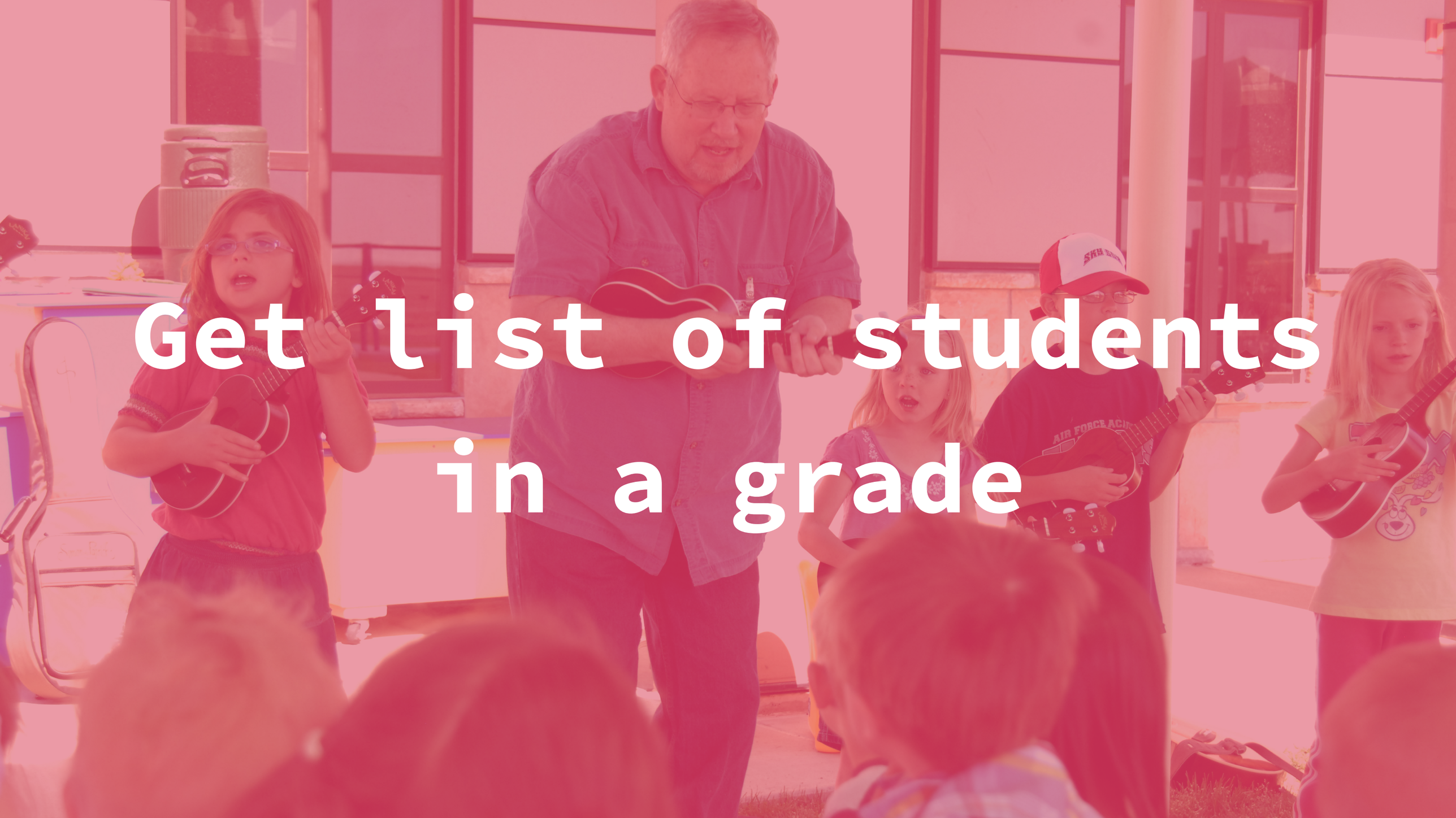
```
def add(name, grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students] << name
  else
    roster << { grade: grade, students: [name] }
  end
end
```

Add Student Name to Roster

```
def add(name, grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students] << name
  else
    roster << { grade: grade, students: [name] }
  end
end
```

Add Student Name to Roster

```
def add(name, grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students] << name
  else
    roster << { grade: grade, students: [name] }
  end
end
```



Get list of students in a grade

Get list of students in a grade

```
def students(grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students].sort
  else
    []
  end
end
```

Get list of students in a grade

```
def students(grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students].sort
  else
    []
  end
end
```

Get list of students in a grade

```
def students(grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students].sort
  else
    []
  end
end
```

Get list of students in a grade

```
def students(grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students].sort
  else
    []
  end
end
```

Get list of students in a grade

```
def students(grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students].sort
  else
    []
  end
end
```

Get list of students in a grade

```
def students(grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students].sort
  else
    []
  end
end
```

Get sorted list of students in all grades

Get sorted list of students in all grades

```
def students_by_grade
  roster
    .sort_by { |enrollment| enrollment[:grade] }
    .map do |enrollment|
      enrollment[:students] =
        enrollment[:students].sort
      enrollment
    end
end
```

Get sorted list of students in all grades

```
def students_by_grade
  roster
    .sort_by { |enrollment| enrollment[:grade] }
    .map do |enrollment|
      enrollment[:students] =
        enrollment[:students].sort
      enrollment
    end
end
```

Get sorted list of students in all grades

```
def students_by_grade
  roster
    .sort_by { |enrollment| enrollment[:grade] }
    .map do |enrollment|
      enrollment[:students] =
        enrollment[:students].sort
      enrollment
    end
end
```

Get sorted list of students in all grades

```
def students_by_grade
  roster
    .sort_by { |enrollment| enrollment[:grade] }
    .map do |enrollment|
      enrollment[:students] =
        enrollment[:students].sort
      enrollment
    end
end
```

Get sorted list of students in all grades

```
def students_by_grade
  roster
    .sort_by { |enrollment| enrollment[:grade] }
    .map do |enrollment|
      enrollment[:students] =
        enrollment[:students].sort
      enrollment
    end
end
```

Get sorted list of students in all grades

```
def students_by_grade
  roster
    .sort_by { |enrollment| enrollment[:grade] }
    .map do |enrollment|
      enrollment[:students] =
        enrollment[:students].sort
      enrollment
    end
end
```

```
exercism submit grade_school.rb
```

Done !

Done?

what gives?

what gives?

-  Methods are kind of long...

what gives?

- Methods are kind of long...
- Is the naming right...?

what gives?

- Methods are kind of long...
- Is the naming right...?
- There's code repetition...

what gives?

- Methods are kind of long...
- Is the naming right...?
- There's code repetition...
- Something doesn't *feel* right...

Take 2

Create a School Roster

```
class School
  def initialize
    @roster = []
  end

  private

  attr_reader :roster
end
```

A Roster is filled with Enrollments

```
> @roster
=> [
      { grade: 1, students: ["Paul", "RobC"] },
      { grade: 2, students: ["Nick"]) },
      { grade: 3, students: ["Cath", "RobH"]) }
   ]
```

Simplify to a Hash of Enrollments

```
> @enrollments
=> {
       1 => ["Paul", "RobC"],
       2 => ["Nick"],
       3 => ["Cath", "RobH"]
   }
```

Create a Hash of Enrollments

```
class School
  def initialize
    @enrollments = {}
  end

  private

  attr_reader :enrollments
end
```

Add Student Name to Roster

```
def add(name, grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students] << name
  else
    roster << { grade: grade, students: [name] }
  end
end
```

Add Student Name to Roster

```
def add(name, grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students] << name
  else
    roster << { grade: grade, students: [name] }
  end
end
```

Add Student Name to Roster

```
def add(name, grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students] << name
  else
    roster << { grade: grade, students: [name] }
  end
end
```

Get list of students in a grade

```
def students(grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students].sort
  else
    []
  end
end
```

Get list of students in a grade

```
def students(grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students].sort
  else
    []
  end
end
```

Get list of students in a grade

```
def students(grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students].sort
  else
    []
  end
end
```


Too much
repetition

Create a Hash of Enrollments

```
class School
  def initialize
    @enrollments = {}
  end

  private

  attr_reader :enrollments
end
```

Create a Hash of Enrollments

```
class School
  def initialize
    @enrollments =
      Hash.new { |hash, key| hash[key] = [] }
  end

  private

  attr_reader :enrollments
end
```

Create a Hash of Enrollments

```
class School
  def initialize
    @enrollments =
      Hash.new { |hash, key| hash[key] = [] }
  end

  private

  attr_reader :enrollments
end
```

Add Student Name to Roster

```
def add(name, grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students] << name
  else
    roster << { grade: grade, students: [name] }
  end
end
```

Add Student Name to Enrollments

```
def add(name, grade)
  enrollments[grade] << name
end
```

Get list of students in a grade

```
def students(grade)
  grade_enrollment =
    roster.find { |enrollment| enrollment[:grade] == grade }
  if grade_enrollment
    grade_enrollment[:students].sort
  else
    []
  end
end
```

Get list of students in a grade

```
def students(grade)
  enrollments[grade].sort
end
```


ROLLMASTR

Get sorted list of students in all grades

```
def students_by_grade
  roster
    .sort_by { |enrollment| enrollment[:grade] }
    .map do |enrollment|
      enrollment[:students] =
        enrollment[:students].sort
      enrollment
    end
end
```

Get sorted list of students in all grades

```
def students_by_grade
  enrollments.sort.map do |grade, students|
    { grade: grade, students: students.sort }
  end
end
```

Get sorted list of students in all grades

```
def students_by_grade
  enrollments.sort.map do |grade, students|
    { grade: grade, students: students.sort }
  end
end
```

Get sorted list of students in all grades

```
def students_by_grade
  enrollments.sort.map do |grade, students|
    { grade: grade, students: students.sort }
  end
end
```

Get sorted list of students in all grades

```
def students_by_grade
  enrollments.sort.map do |grade, students|
    { grade: grade, students: students.sort }
  end
end
```

Get sorted list of students in all grades

```
> school.students_by_grade
=> [
      { grade: 1, students: ["Paul", "RobC"] },
      { grade: 2, students: ["Nick"]) },
      { grade: 3, students: ["Cath", "RobH"]) }
   ]
```

You'll rarely get it right the first time

Submit often, get feedback

Read other people's code

Refactor until you're happy

github.com/paulfioravanti/exercism

Thanks!

@paulfioravanti