

OBJECT-ORIENTED PROGRAMMING - EXERCISE 1

You should use `rspec` to test your classes, and use `irb` to load your file and debug them manually.

Question 1. Write a new `Person` class that implements the following interface:

```
person = Person.new("joe", "bloggs", "1 Jan 1990") # The date should be optional
person.dob # => A Date object
person.first_name # => "Joe"
person.surname # => "Bloggs"
person.fullname # => "Joe Bloggs"

person.emails # => []
person.add_email "joe@foo.com"
person.add_email "joe.bloggs@work.com"

person.emails # => ["joe@foo.com", "joe.bloggs@work.com"]

person.phone_numbers # => []
person.add_phone "07712345678"
person.add_phone "02012345678"

person.phone_numbers # => ["07712345678", "02012345678"]

person.remove_email(0)
person.emails # => ["joe.bloggs@work.com"]

person.to_s
# => "Joe Bloggs was born on 1990-01-01.\n Their email addresses are:
[\"joe@foo.com\"].\n Their phone numbers are [\"07712345678\", \"02012345678\"]"
```

Additional Constraints:

- Emails should be stored in an array, so should phone numbers.
- It should not be possible to change the emails or phone numbers arrays except using the `add_email`, `add_phone`, `remove_email` and `remove_phone` methods.
- The Array method `delete_at` might be useful.

Question 2. Add a `print_details` method that prints out the details of the person, and then lists their emails and phone numbers on separate lines. You should use the Array method `each`. The `strftime` Date method might be useful. **You should only try to write tests for this once you have finished everything else.** Here's a suggested output format:

```
Joe Bloggs
-----
Date of Birth: 01 January 1990

Email Addresses:
- joe@foo.com
- joe.bloggs@work.com

Phone Numbers:
- 02012345678
```

Question 3. Add a `FamilyMember` class that inherits from `Person` and has an additional relationship instance variable.

Question 4. Write an `AddressBook` class that implements the following interface:

```
person1 = Person.new("joe", "blogs", "1 Jan 1990")
person2 = Person.new("andy", "nother", "2 Jan 1995")
book = AddressBook.new
book.add person1
book.add person2
```

```
book.list # Should print this to the command line:
```

```
Address Book
```

```
-----
```

```
Entry 1: Joe Bloggs
```

```
Entry 2: Andy Nother
```

Additional Constraints:

- Emails should be stored in an array, so should phone numbers.
- It should not be possible to change the emails or phone numbers arrays except using the `add_email`, `add_phone`, `remove_email` and `remove_phone` methods.
- The Array method `each_with_index` might be useful.

Bonus points:

- Throw an error if the `add` method is passed anything but a `Person` or `FamilyMember` object.