Ruby

What is Ruby

- Language for scripting and applications
- Object Oriented
- General purpose

- irb
- ruby.exe
- Notepad
- RubyMine

Java

```
public class SomeClass {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Ruby

```
puts "Hello World"
```

Java

```
public class SomeClass {
    private int age;
    public void setAge(int age) {
        this.age = age;
    }
    public int getAge() { return age; }
}
```

Ruby

```
class SomeClass
    attr_accessor :age
end
```

With Accessors

```
class SomeClass
    attr_accessor :age
end
```

Without Accessors

```
class SomeClass
    def age=(age)
        @age = age
    end
    def age
        @age
    end
end
```

Java

```
public class SomeClass {
    private int age;
    public SomeClass(int age) {
        this.age = age;
    }
}
```

Ruby

```
class SomeClass
    attr_accessor :age
    def initialize(age)
        @age = age
    end
end
```

Scope

- `Global - $some_var`
- `Instance - @some_var`
- `Class - @@some_var`
- `Local – some_var`

Strings

- Immutable – 'some string'
- Interpolated – "some string"
- Symbols - :some_string

Java

```
public class Horse extends Animal {
…
}
```

Ruby

```
class Horse < Animal
…
end
```

Java

```
for(String name : names) {
    System.out.println(name);
}
```

Ruby

```
names.each do |name|
    puts name
end
```

Ruby Iterators

```
names.each do |name|
    puts name
end

rev_names = names.collect do |name|
    name.reverse
end

long_names = names.select do |name|
    name.size > 25
end

first_name_with_z = names.find do |name|
    name.include? 'z'
```

Java

```
import org.junit.*;
public class HorseTests {
    @Test
    public void eatingLotsCausesWeightGain() {
        assertTrue(…);
    }
```

Ruby

```
require 'rspec'
describe 'Horse' do
    it 'gains weight when it eats a lot' do
        expect(…).to be_true
    end
end
```

Java

```
int anArray = new int[10];
anArray[0] = 1;
anArray[4] = 7;

//Can't change the size
//Alternative is ArrayList
```

Ruby

```
anArray = [] # or Array.new
anArray << 1
anArray << 5

other = [1,2,3,4]
```

Java

```
Map map = new HashMap();
map.put("a", 1);
map.put("b", 2);

map.get("b");
```

Ruby

```
map = {} # or Hash.new
map[:a] = 1
map[:b] = 2

map[:b]

other  = {a: 1, b: 2, c: 3}
```

Other Notes

- Case conventions
  - Camel case for class names
  - Snake case for variables and methods
  - Upper case for constants
- No relationship between file name and class name
- Interpreted, not compiled
- Can write scripts or applications
- Built-in REPL called IRB
- Single inheritance
  - No interfaces
  - Uses mix-ins
- Return statements not required
- Parenthesis not required (except when they are!!!!)

Other Notes

- Everything is an object
  - You can invoke methods on things like numbers and string
- nil vs. null
- Special characters are permitted in method names
  - my_object.nil?
  - my_Object.some_property = 7

Labs

- Intent
  - Learn ruby and rspec
  - Create tests that are also documentation
  - Practice good OO programming
- Labs
  - Calculator
  - Poker Hands