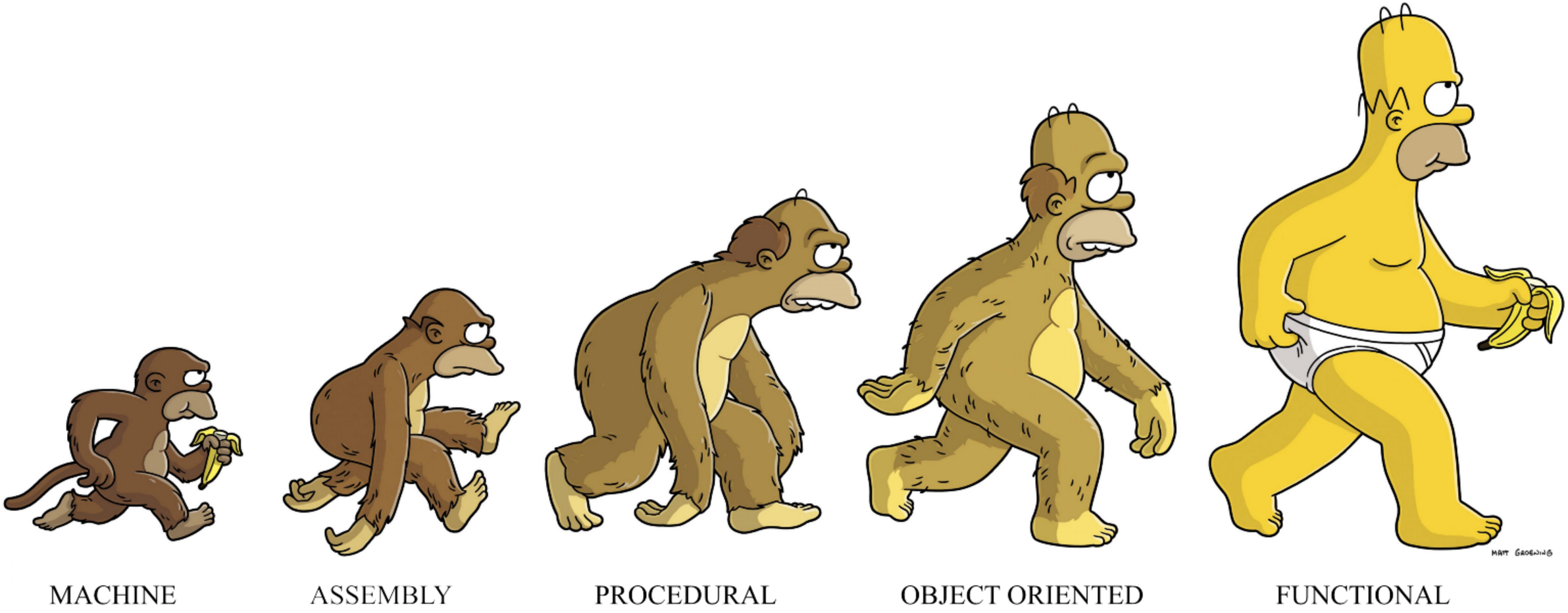

SPRING FRAMEWORK
GURU

What is Functional Programming?



Credit - Charles Scalfani

Imperative Programming

- “In computer science, imperative programming is a programming paradigm that uses statements that change a program's state. In much the same way that the imperative mood in natural languages expresses commands, an imperative program consists of commands for the computer to perform.” Source: Wikipedia

Imperative Example

```java
@Test
public void countDogsWithEightCharactersImpd() throws Exception {
    /*
    Get count of dogs with 6 characters in name
     */

    List<String> dogs = Arrays.asList("Vizsla", "Lab", "Golden", "GSP", "Poodle", "Yorkie", "Mutt");

    int dogCount = 0;

    for (String dog : dogs) {
        if (dog.length() == 6) {
            dogCount++;
        }
    }

    System.out.println(dogCount);
}
```

Functional Programming

- “In computer science, functional programming is a programming paradigm - a style of building the structure and elements of computer programs - that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data.” Source: Wikipedia

Functional Example

```
@Test
public void countDogsWithEightCharactersDecd() throws Exception {
    /*
    Get count of dogs with 6 characters in name
     */

    List<String> dogs = Arrays.asList("Vizsla", "Lab", "Golden", "GSP", "Poodle", "Yorkie", "Mutt");

    System.out.println(dogs
            .stream()
            .filter(dog -> dog.length() == 6)
            .collect(Collectors.toList())
            .size());
}
```

Imperative vs Declarative

Imperative	Declarative
How to do it	What to do
Mutable	Immutable (Transforms)
Has side effects	No side effects
Pass Objects	Can also pass functions
Hard to Compose	Functional Composition
Not Threadsafe	Threadsafe

Mutability

- Mutability are objects that can change
- Immutable are objects that do not change

```
int dogCount = 0;

for (String dog : dogs) {
    if (dog.length() == 6) {
        dogCount++;
    }
}
```

Mutability vs Immutable

- Mutable objects are error prone and hard to understand
- Immutable objects are easier to use
- Immutable objects are thread safe
- Mutable objects open the door to concurrency problems

Final variables in Java

- Final variables in Java can still be mutated
- Once initialized, the variable cannot be re-assigned
- BUT - state of the object can change, if properties are not final

Use of final in Spring

```java
@Service
public class MovieServiceImpl implements MovieService {

    private final MovieRepository movieRepository;

    public MovieServiceImpl(MovieRepository movieRepository) {
        this.movieRepository = movieRepository;
    }
```

Pure Functional vs Functional Style

- Pure Functional Languages will not allow any mutability
  - Haskel is an example
- Functional Style languages will encourage immutability
- BUT immutability is not strictly enforced
  - Java is a Functional Style Language

Functions

- Can be passed objects
- Can create objects
- Can return objects

Higher Order Functions

- Can be passed objects and functions
- Can create objects and functions
- Can return objects and functions
- Java supports higher order functions


SPRING FRAMEWORK
GURU