

# Object Oriented Caesar Cipher

## Introduction

# Object Oriented Programming

- Java: Object Oriented Language
  - What does that mean?



Java™

# Object Oriented Programming

- Java: Object Oriented Language
  - What does that mean?
- Encapsulate code and data

Object = Code + Data

# Object Oriented Programming

- Java: Object Oriented Language
  - What does that mean?
- Encapsulate code and data

Object = Code + Data

Methods Fields

```
graph TD; Object[Object] -- "=" --- Code[Code]; Object -- "+" --- Data[Data]; Code --- Methods[Methods]; Data --- Fields[Fields];
```

# Familiar: String, ImageResource, ...

- Familiar example: String
  - Data: sequence of characters
  - Code: indexOf, substring, ...

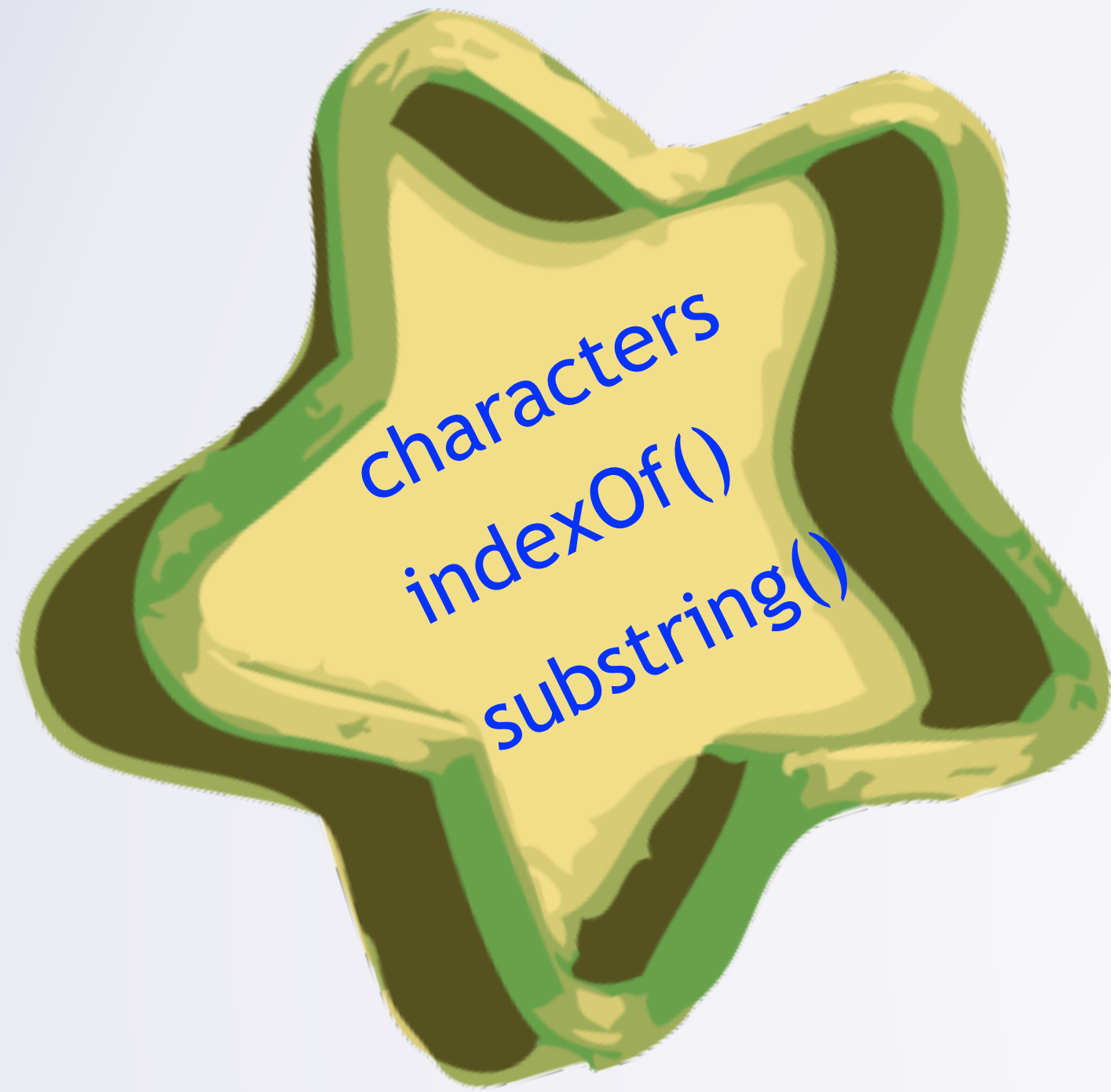
**CTACGATGCCTGATGA**

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5

| method                            | value           |
|-----------------------------------|-----------------|
| <code>s.length()</code>           | 15              |
| <code>s.indexOf("program")</code> | 4               |
| <code>s.indexOf("q")</code>       | 7               |
| <code>s.indexOf("f")</code>       | -1              |
| <code>s.indexOf("q", 8)</code>    | 14              |
| <code>s.startsWith("duke")</code> | true            |
| <code>s.endsWith("king")</code>   | false           |
| <code>s.substring(4, 7)</code>    | "pro"           |
| <code>s.toUpperCase()</code>      | DUKEPROGRAMMING |



# Class = Type, Object = Instance



- Classes are types:
  - Define what is in objects of that type
- Objects are instances of class
  - Can make many (use: `new`)

# Class = Type, Object = Instance



```
characters = H e l l o  
int indexOf(String str) {...}  
String substring(int beginIndex) {...}
```

```
characters = W o r l d  
int indexOf(String str) {...}  
String substring(int beginIndex) {...}
```

- Classes are types:
  - Define what is in objects of that type
- Objects are instances of class
  - Can make many (use: `new`)



# Why OO?



- Why objects and classes?
  - Facilitate large programs
- Some basics here: **fields, constructors, visibility**
- Java Programming: Principles of Software Design