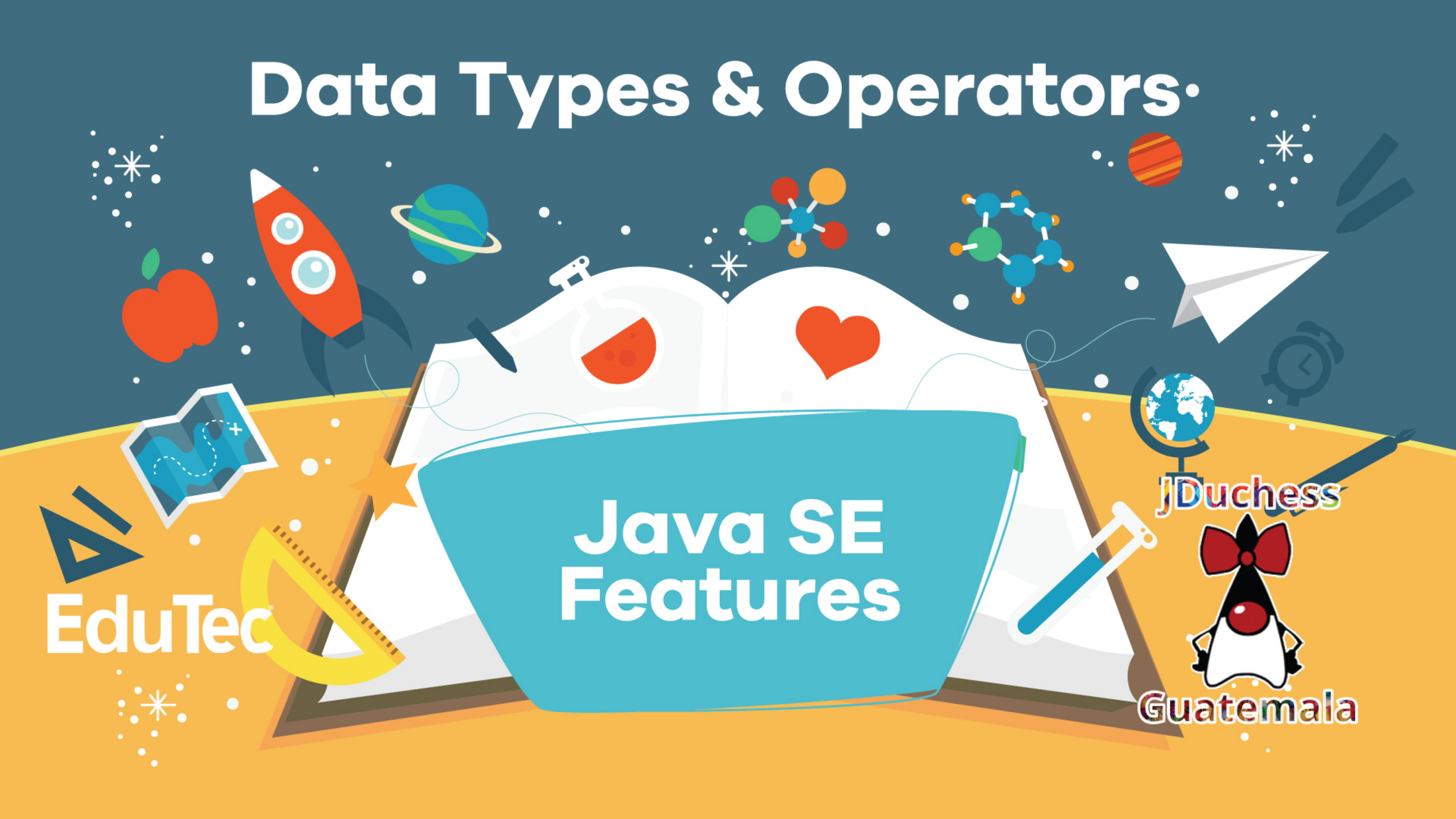
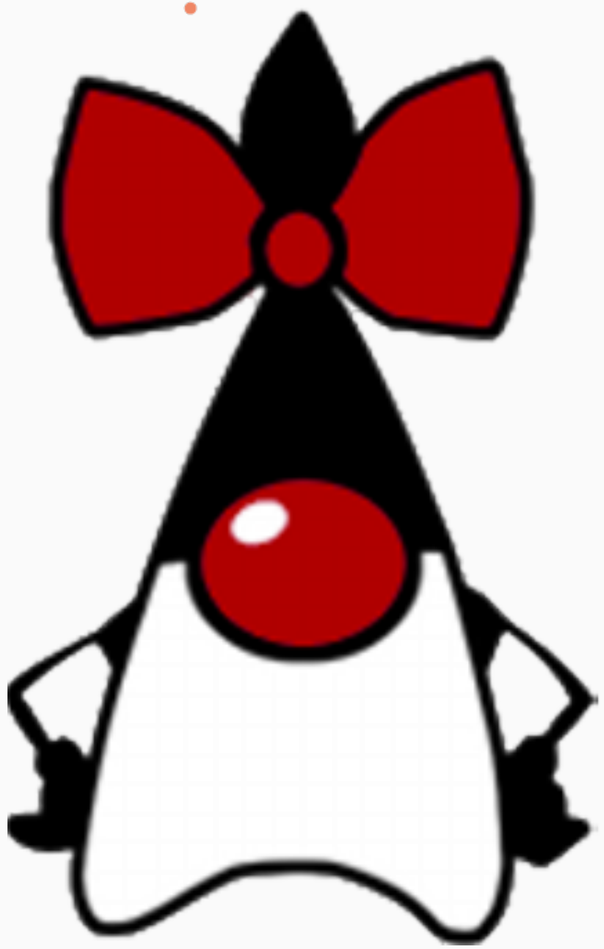
Data Types & Operators





JDuchess Guatemala

@itrjwyss

AMBASSADOR

Mercedes Wyss

@itrjwyss

Community Leader

Devs+502 & JDuchess Chapter Guatemala

Ex-JUG Member

Guatemala Java Users Group (GuateJUG)

Chief Technology Officer (CTO) at Produactivity

Full Stack Developer

Auth0 Ambassador &

Oracle Groundbreaker Ambassador

Duke's Choice Award

2016 Winner

Java™

AMBASSADOR

Oracle Groundbreaker Ambassador

Devs +502

Comunidad Desarrolladores en Tecnologías

< Google > en Guatemala

JDuchess

Guatemala

Roadmap

@itrjwyss

Oracle
Groundbreaker
Ambassador

Java SE 8 - OCA

- Java Basics
- Understanding The OOP in Java
- Core Java APIs and Complex Features
- Disruptive Features

@itrjwyss

Java Basics

- Java Platform and Main Structure
- Data Types, Variables and Operators
- Decision and Loop Constructs
- Advanced Data Types

@itrjwyss

Oracle
Groundbreaker
Ambassador

Understanding The OOP in Java

- OOP and Packages in Java
- Methods, variable Scope and Encapsulation
- Inheritance
- Polymorphism

@itrjwyss

Core Java APIs and Complex Features

- Abstract Classes and Interfaces
- String and StringBuilder Objects
- Arrays and ArrayList
- Handling Exceptions

Disruptive Features

- Date and Time APIs
- Functional Programming (Lambdas)

Java Basics

@itrjwyss

Oracle
Groundbreaker
Ambassador

Java Platform and Main Structure

- Java Platform
- Java Class Structure
    - Structure & Signature
    - Naming conventions
- Compiling and Interpreting Java Code

@itrjwyss

Data Types, Variables and Operators

- Data Types (primitive types, space in memory)
- Declaring and Initializing Variables (assignment statement)
- Understanding Operators ([arithmetic, numerical, relational, logical, equality, etc], operator precedence)

Decision and Loop Constructs

- Conditional Statements (if, if-then, if-then-else, ternary operator and switch)
- Iteration Statements (for, while, and do-while)
- Transfer of Control Statements (break, continue, return and labeled)

Advanced Data Types

- Wrapper Classes and Autoboxing (primitive data types as objects)
- Numeric Promotion
- Enumerations

Understanding the OOP in Java

@itrjwyss

OOP and Packages in Java

- Difference and Relationship between Class and Object
- The Pillars of the OOP (encapsulation, inheritance and polymorphism)
- Java Access and Non-access Modifiers (public, private, protected, default, static, final, abstract)
- Java Packages

@itrjwyss

Oracle
Groundbreaker
Ambassador

Methods, Variable Scope and Encapsulation

- Method Structure
- Variable Scope (Default Initialization, apply access modifiers)
- The Encapsulation Principle
- Constructors (overloading, object's lifecycle, super, this)

Inheritance

- The Inheritance Principle
- The Impact of Access Modifiers in Inheritance
- Overriding

Polymorphism

- The Polymorphism Principle
- Casting
- Virtual Methods and Polymorphic Parameters

Core Java APIs and Complex Features

@itrjwyss

Oracle Groundbreaker Ambassador

Abstract Classes and Interfaces

- Abstract Classes
- Interfaces
- Abstract Classes vs Interfaces
- Implementing Polymorphism

@itrjwyss

String and StringBuilder Objects

- The String Object (create and manipulate Strings | Understand immutability)
- StringBuilder
  - Manipulate data using the StringBuilder class and its methods
  - Understand mutability
  - Discover StringBuffer class
- Comparing Strings and other Objects (using == or equals() or compareTo() methods)

Arrays and ArrayList

- One-dimensional Arrays
- Operations on Arrays
- Multi-dimensional Arrays
- ArrayList
- Operations on ArrayList

@itrjwyss

Handling Exceptions

- Understanding Exceptions
- Exception Flow
- Methods that Throw Exceptions
- Recognizing Common Exceptions

Disruptive Features

@itrjwyss

Date and Time APIs

- JSR 310 Overview (java.time)
- Date and Time Creation and Manipulation
- Period and Duration Objects
- Formatting and Parsing Date and Times

Functional Programming (Lambdas)

- Functional Programming Fundamentals
- Lambdas and Functional Interfaces
- Using Predicate

Mock Tests

@itrjwyss

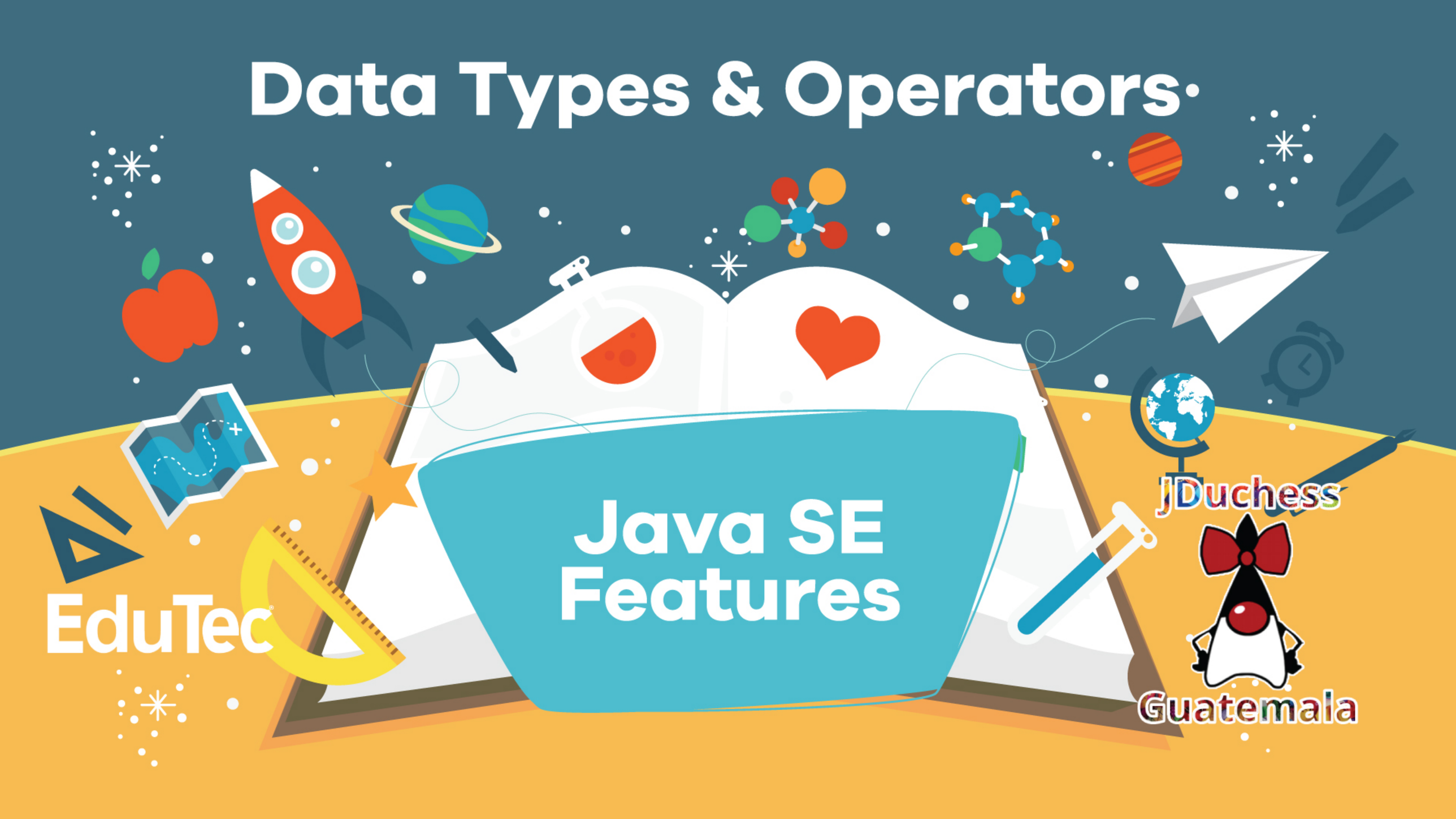
Oracle
Groundbreaker
Ambassador

https://forms.gle/yrUQQJf91QyKdH349

http://bit.ly/32MTbaq

@itrjwyss

Data Types & Operators



Data Types, Variables and Operators

- Data Types (primitive types, space in memory)
- Declaring and Initializing Variables (assignment statement)
- Understanding Operators ([arithmetic, numerical, relational, logical, equality, etc], operator precedence)

Variable

- Es el nombre de una pieza de memoria que almacena información.
- Son la piedra angular para almacenar información en un programa.

Variables Primitivas

- Es la forma más básica de almacenar información.
- Cuando una variable primitiva es declarada, se reserva memoria para almacenar su valor.
- Estos valores son almacenados en el Stack Memory junto a las referencias a objetos.

Identificadores

- Deben empezar con una letra o el símbolo $ o _
- Después de caracteres pueden haber números
- No se puede utilizar ninguna de las palabras reservadas de Java

Palabras Reservadas

abstract	class	extends	implements	null	strictfp	true
assert	const	false	import	package	super	try
boolean	continue	final	instanceof	private	switch	void
break	default	finally	int	protected	synchronized	volatile
byte	do	float	interface	public	this	while
case	double	for	long	return	throw	
catch	else	goto	native	short	throws	
char	enum	if	bew	static	transient	

Naming Conventions

- Se definen con Lower CamelCase, la primera palabra siempre empieza con minúscula, en palabras compuestas desde la segunda palabra se empieza con mayúscula.

- Es permitido utilizar los caracteres dollar ($) y guión bajo (_), incluso al inicio del nombre.

- Lo más importantes es utilizar nombres mnemónicos (representen lo que la variable almacena)

- Por ejemplo
  meetupDate, meetupPlace, $startDollar, name3

Tipos de Datos Primitivos

- boolean
- char (character)
- byte
- short
- int (Integer)
- long
- float (floating point)
- double (double precision floating point)

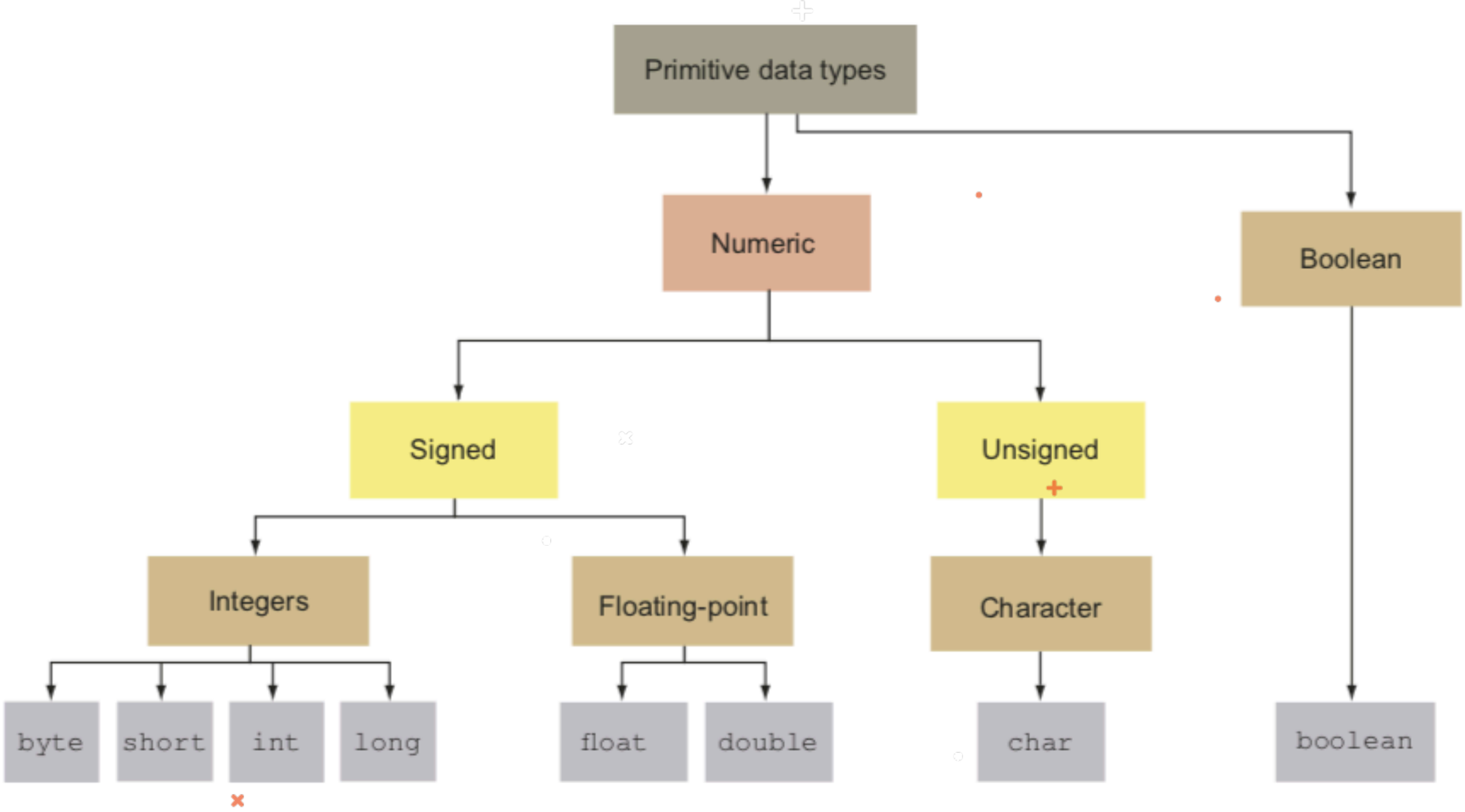


Figure 2.2 Categorization of primitive data types

Espacio en Memoria

Data Type	Used for	Size	Range
`boolean`	`true` or `false`	1 bit	N/A
`char`	Unicode character	16 bits	\u0000 to \uFFFF (0 to 65,535)
`byte`	integer	8 bits	–128 to 127
`short`	integer	16 bits	–32768 to 32767
`int`	integer	32 bits	–2,147,483,648 to 2,147,483,647
`long`	integer	64 bits	-2^{63} to $2^{63}-1$
`float`	floating point	32 bits	positive $1.4e^{-45}$ to $3.4e^{+38}$
`double`	floating point	64 bits	positive $5e^{-324}$ to $1.8e^{+308}$

Rangos

Table 2.4 Ranges of values stored by the signed numeric Java primitive data types

Data type	Size	Range of values
`byte`	8 bits	–128 to 127, inclusive
`short`	16 bits	–32,768 to 32,767, inclusive
`int`	32 bits	–2,147,483,648 to 2,147,483,647, inclusive
`long`	64 bits	–9,223,372,036,854,775,808 to 9,223,372,036,854,775,807, inclusive

Rangos

Table 2.6 Range of values for decimal numbers

Data type	Size	Range of values
`float`	32 bits	+/–1.4E–45 to +/–3.4028235E+38, +/–infinity, +/–0, NaN
`double`	64 bits	+/–4.9E–324 to +/–1.7976931348623157E+308, +/–infinity, +/–0, NaN

Assignment Statement

- Una declaración de asignación establece un valor dentro de una variable.
- Son parte de las expression statements.
- = += -= *= /= (primero (+, -, *, /) después asigna)

```
variable = value
```


Declaración

```
String s1;
String s2, s3, s4;

int num, String value;
```


Declaración y Asignación

```
int fishInTank = 100;
int fishInCooler = 50;
```

@itrjwyss

AMBASSADOR

Declaración

```
String s1;
String s2, s3, s4;

int num, String value;
```


Declaración y Asignación

```
String s3 = "yes", s4 = “no";

int i1, i2, i3 = 4;

char c1 = '\u0122';
char c2 = 122;
char c3 = 'z';
```


Declaración y Asignación

```
String s3 = "yes", s4 = “no";

int i1, i2, i3 = 4;

char c1 = '\u0122';
char c2 = 122;
char c3 = 'z';
```

Es la única variable inicializada

@itrjwyss

Es necesario inicializarlos

```
public void notValid() {
    int y = 10;
    int x;
    // DOES NOT COMPILE
    int reply = x + y;
}
```

https://github.com/itrjwyss/

https://www.facebook.com/itrjwyss

@itrjwyss