OOP PRESENTED BY ALEEZA….

OBJECT ORIENTED PROGRAMMING(OOP)

- OOP in JavaScript stands for Object-Oriented Programming.
- It's a programming paradigm that focuses on organizing code

into objects that have both data (properties) and behavior (methods).

- Object-Oriented Programming (OOP) in JavaScript is a way of

writing code where we organize our data (like variables)

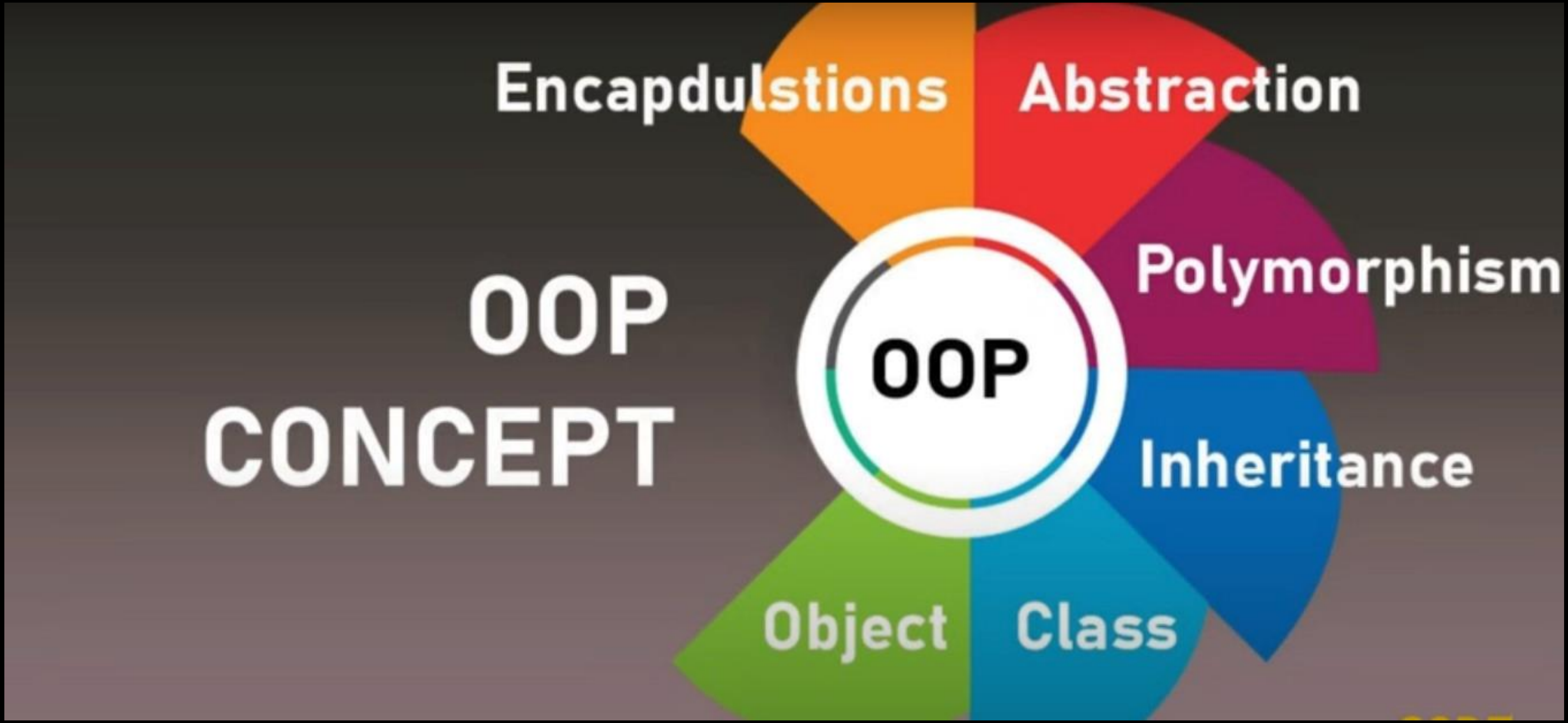
and functions (like actions we want to do with the data) into bundles called "objects".

- These objects can interact with each other,

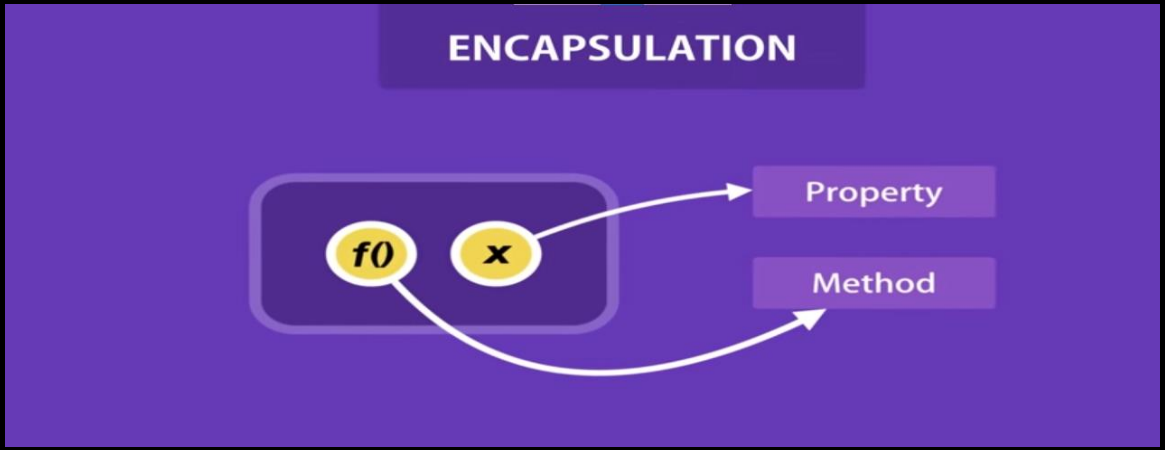
and each object can have its own unique set of data and functions.

- OOP helps make our code easier to manage and reuse.

Object-Oriented Programming (OOP) in JavaScript can be understood through the four main pillars or principles of OOP. These principles are:

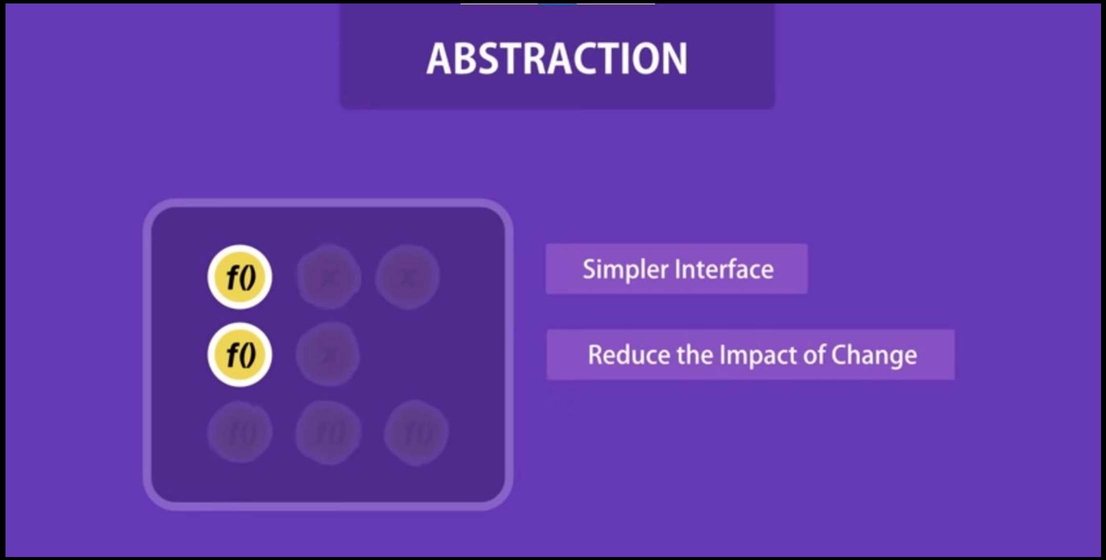


1) Encapsulation:



Encapsulation is the bundling of data (properties) and methods (functions) that operate on the data into a single unit (class or object).
Implementation in JavaScript: In JavaScript, encapsulation can be achieved through closures and object literals.
Private variables and methods can be simulated using closures,
while public methods can access and modify these
private variables.

2) Abstraction:



Abstraction involves hiding the complex implementation
details of an object and exposing only the essential features
or functionalities.
Implementation in JavaScript: Abstraction in JavaScript can be achieved by defining interfaces
(public methods)
that provide access to the object’s functionality while hiding its underlying implementation.

3) Inheritance:

```
Parent Class
      |
      |
      v
 Child Class
```

Inheritance allows one object (subclass/derived class) to acquire properties and behaviors of another object (superclass/base class).

It promotes code reusability and supports hierarchical classification.

Implementation in JavaScript: JavaScript uses prototypal inheritance, where objects can inherit properties and methods from other objects. This is achieved by linking objects through prototypes.

Prototype inheritance in javascript is the linking of prototypes a parent object to a child object to share and utilize the properties of a parent class using a child class. Prototypes are hidden objects that are used to share the properties and methods of a parent class with child classes.

4) Polymorphism:

Polymorphism allows objects to be treated as instances of their parent class, but still behave like their own class.

It enables flexibility and dynamic behavior in OOP.

Implementation in JavaScript: Polymorphism in JavaScript can be achieved through method overriding. Subclasses can override methods of their superclass to provide specific implementations while still adhering to a common interface.

These pillars of OOP provide a structured approach to designing
and implementing software systems in JavaScript,promoting modularity, reusability, and
maintainability of code.
Understanding and applying
these principles
effectively can lead to better-designed applications .

THANK YOU …..