ES6 Features

1.LET , VAR , CONST

The `let` keyword allows you to declare a variable with block scope.

The const keyword allows you to declare a constant

Constants are similar to let variables, except that the value cannot be changed.

var declarations are globally scoped or function scoped while let and const are block scoped. var variables can be updated and re-declared within its scope; let variables can be updated but not re-declared; const variables can neither be updated nor re-declared

2.CLASS

Class are the templates or blueprint for the objets created.
classes are the special type of functions. We can define the class just like function declarations and function expressions.

3.OBJECTS

A javaScript object is an instance of class that entity having state and behavior (properties and method).

4.INHERITENCE

The JavaScript inheritance is a mechanism that allows us to create new classes on the basis of already existing classes.exixting class is called super class and new class is sub class.

We use extends keyword to inherit the properties of existings class to the new class
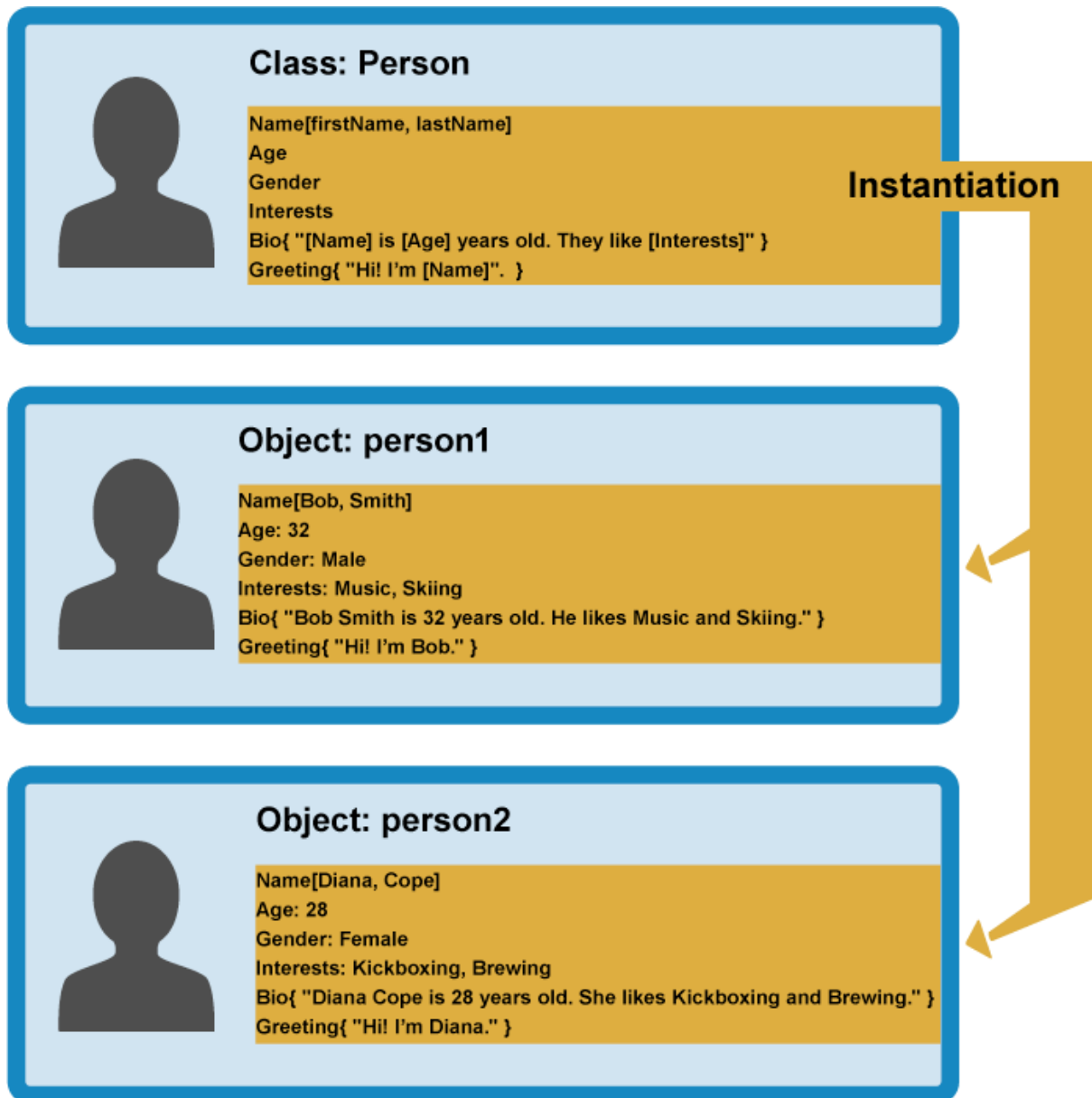
5.CONSTRUCTOR

 JavaScript constructor method is a special type of method which is used to initialize and create an object.

The constructor keyword is used to declare a constructor method.

6.DELEGATION

Delegation means that you use an object of another class as an instance variable, and forward messages to the instance.
Passing reference of the object created.

Class: Person

Name[firstName, lastName]
Age
Gender
Interests
Bio{ "[Name] is [Age] years old. They like [Interests]" }
Greeting{ "Hi! I'm [Name]". }

Instantiation

Object: person1

Name[Bob, Smith]
Age: 32
Gender: Male
Interests: Music, Skiing
Bio{ "Bob Smith is 32 years old. He likes Music and Skiing." }
Greeting{ "Hi! I'm Bob." }

Object: person2

Name[Diana, Cope]
Age: 28
Gender: Female
Interests: Kickboxing, Brewing
Bio{ "Diana Cope is 28 years old. She likes Kickboxing and Brewing." }
Greeting{ "Hi! I'm Diana." }

The ability of multiple object types to implement the same functionality is polymorphism.

SAMPLE EXAMPLE:

INHERITENCE EXAMPLE:

```
class Area {
    constructor(base,height){
        this.base = base
        this.height = height

    }
```

```javascript
triangleArea(){
  let result=(this.base*this.height)/2
  console.log(`Area of Triangle is ${result}`);


}
}

class Perimeter extends Area  {

    constructor(base,height,length,breadth){
       super(base,height)
           this.length=length
           this.breadth=breadth


    }
rectanglePerimeter(){
    let resultRect= this.length*this.breadth;
     console.log(`Perimeter of rectangle is ${resultRect} `)
}
}

  let obj = new Perimeter(10,20,30,40);
  obj.triangleArea();
  obj.rectanglePerimeter();
```

LET VAR CONST EXAMPLE:


```javascript
let city = "USA";

{
  let name = "india";
  console.log(name); // "india";
  const area="germany";
  console.log(area);//"germany"
}

console.log(city); // "USA"
```


EXAMPLE FOR CLASS AND OBJECT:

```javascript
class Car {
    constructor(name, year) {
      this.name = name;
      this.year = year;
    }
  }
```

```
myCar = new Car("Ford", 2014);
console.log( myCar.name + " " + myCar.year)
```