

Introduction To Object Orianted Programming.

Presented By :
Rachita C
B122

Object Oriented Programming

Object-oriented programming is a philosophy or technique for computer programming design that organizes/models software architecture around data or objects rather than functions and logic.

A data field with distinct features and behaviour is referred to as an object.

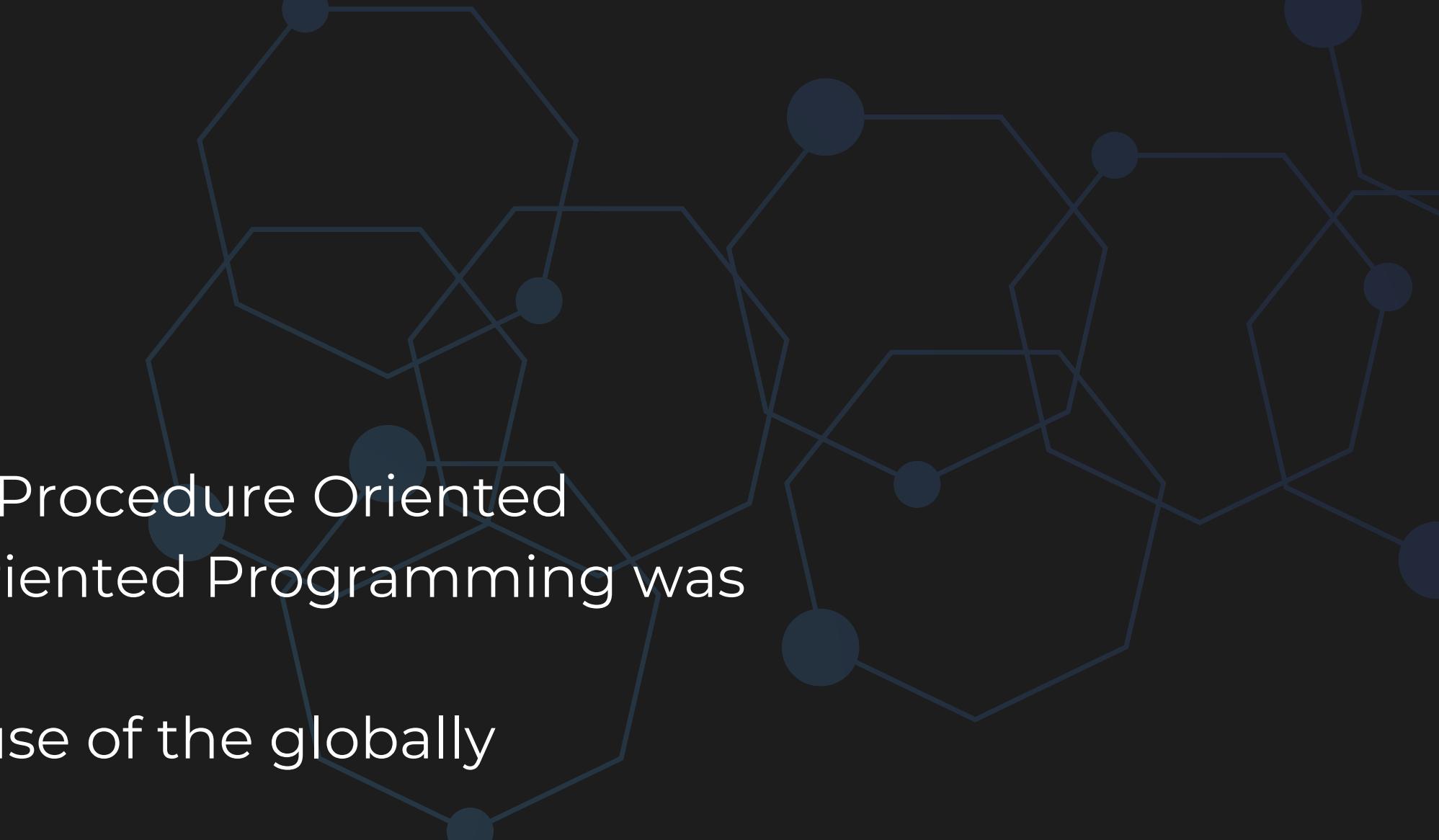
```
function createPerson() {
  const obj = {};
  obj.name = name;
  obj.greeting = function () {
    alert('Hi! I am ' + this.name);
  };
  return obj;
}

newPerson = createP
```

Why OOP Concept was introduced?

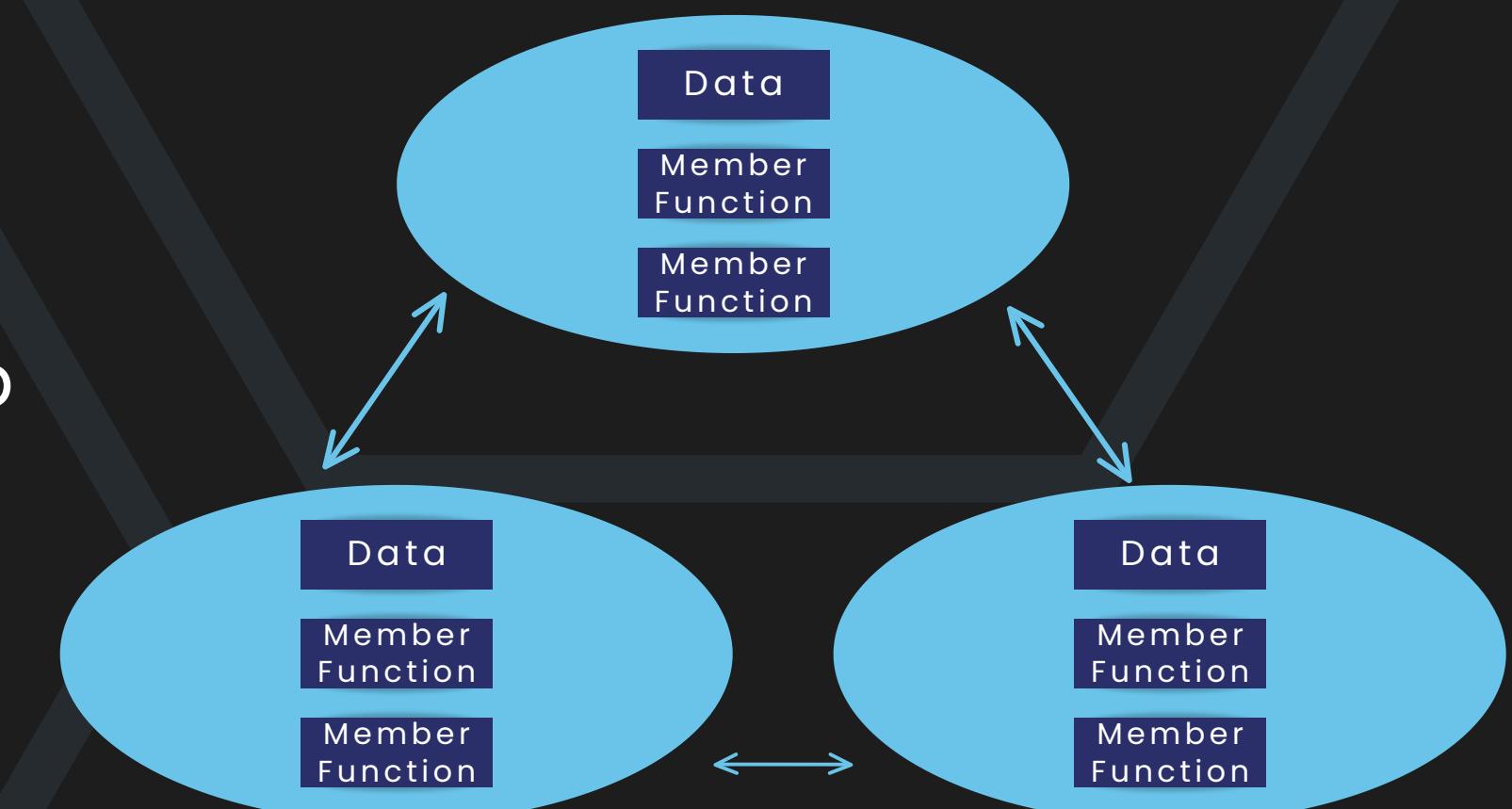
The previous techniques or approach to Procedure Oriented Programming had limits, thus Object-Oriented Programming was created.

- Anyone could corrupt the data because of the globally accessible variables used.
- It is quite difficult to determine which data is utilised by which function in a huge software.
- Does not accurately represent a real-world situation.
- If new data is added, all of the functions that access the data must be changed.



Characteristics

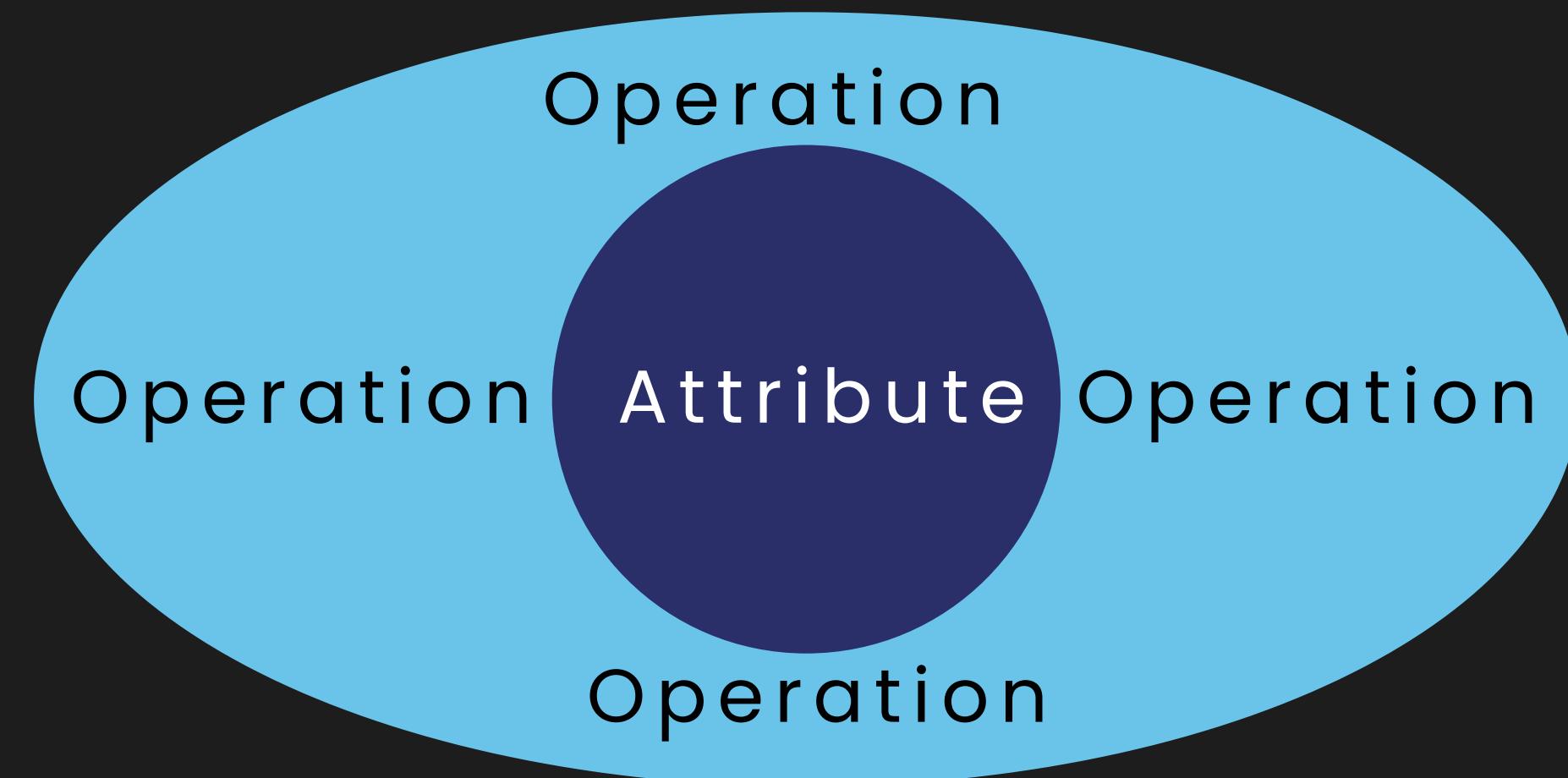
- Programs are divided into entities called object.
- The data structure characterize the objects. And the functions tied to data structure which operate on data.
- Data is hidden and can't accessed externally.
- The objects communicate with each other through functions.
- New data can be added at any point easily.
- The program design used is Bottom up design: Lower level tasks are completed initially, and then merged to offer a solution for a single programme. It promotes code reuse and allow unit testing.



Basic concepts of OOPs

Object:

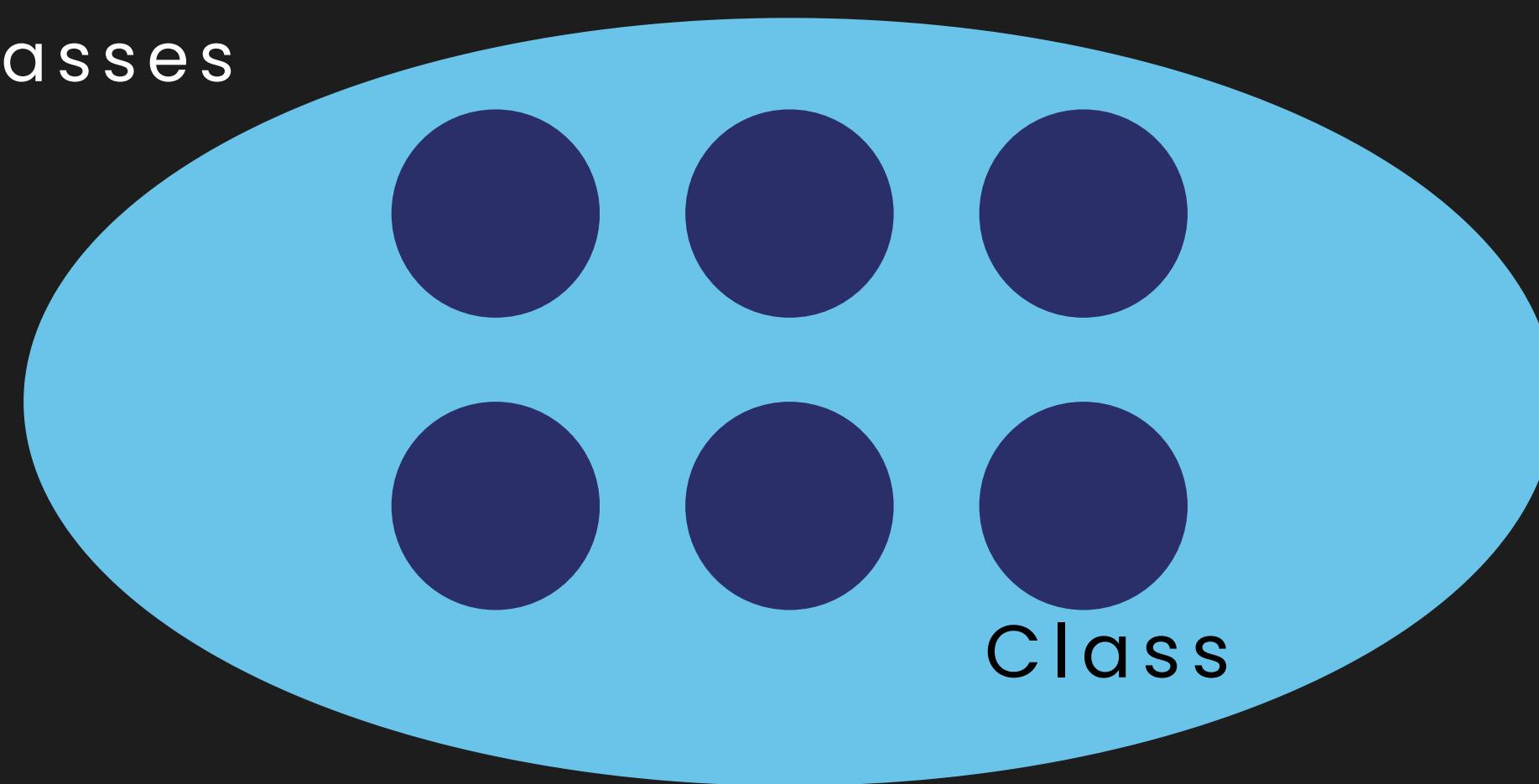
Objects are the basic run time entities of Object orianted programming.They represent Place, person or an item.



Class:

Class is just the blueprint of the datastructure , how it looks. But we dont put any concrete infomation.

Classes



```
class SoftwareEngineer:  
    pass
```

This is a valid class in python we then create an instance of this class.

Instance:

In instance we put all information like age, name , level etc

Difference between class and instance is that Class is just the blueprint of the datastructure , how it looks. But we dont put any concrete infomation but in instance we put all the concrete information.

Method:

The method is a function that is associated with an object. Any object can have its method around it and is not unique to class instance.

Principles of OOPs

- Inheritance
- Polymorphism
- Encapsulation
- Abstraction

Inheritance:

Inheritance is the process by which one class takes on the attributes and methods of another. Newly formed classes are called child classes, and the classes that child classes are derived from are called parent classes. Child classes inherit all of the parent's attributes and methods but can also extend and overrides attributes and methods that are unique to themselves

Polymorphism(Many shapes):

We can write a code that works on the superclass, and it will work with any subclass type as well. Gives a way to use a class exactly like its parent, but each child class keeps its own methods as they are.

Encapsulation:

Encapsulation is the mechanism of hiding of data implementation. Instance variables are kept private and accessor methods are made public to achieve this. With this, we restrict access to public methods (getter/setter). Instance methods can also be kept private.

Abstraction:

Abstraction can be thought of as a natural extension of encapsulation. Applying abstraction means that each object should only expose a high-level mechanism for using it. This mechanism should hide internal implementation details. It should only reveal operations relevant for the other objects.

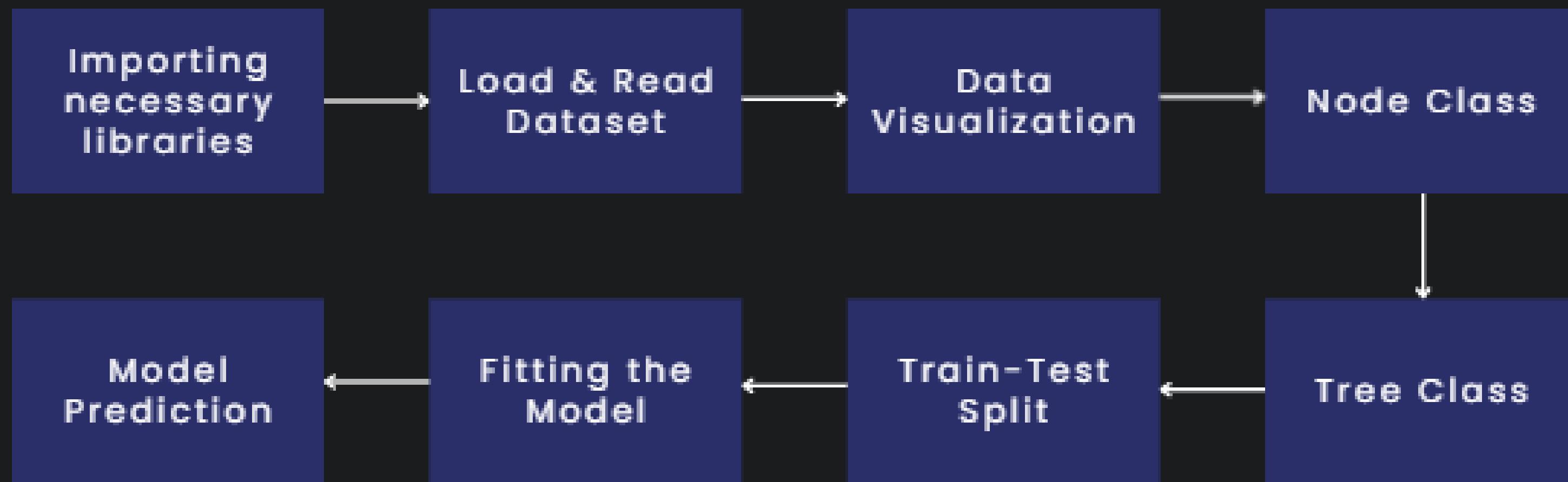
Advantages of OOPs

- Productivity of software development increased
- Polymorphism
- Software maintenance improved
- Quicker improvement
- Cost of development lowered
- Modifying is easy, the data can be added at any point
- Security of the data is maintained
- Good quality software

Disadvantages of OOPs

- Bigger program size
- More slow projects
- Not appropriate for a wide range of issues

Implementation Of Decision Tree





Thank you.