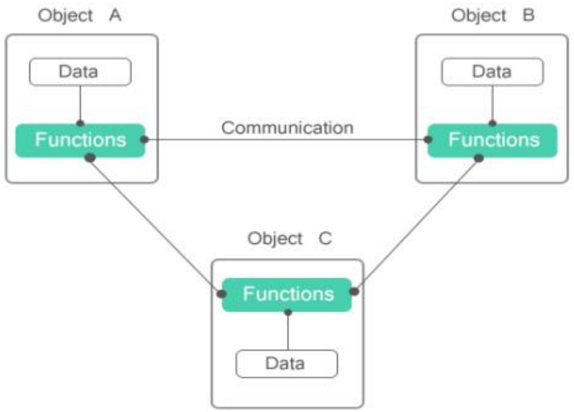
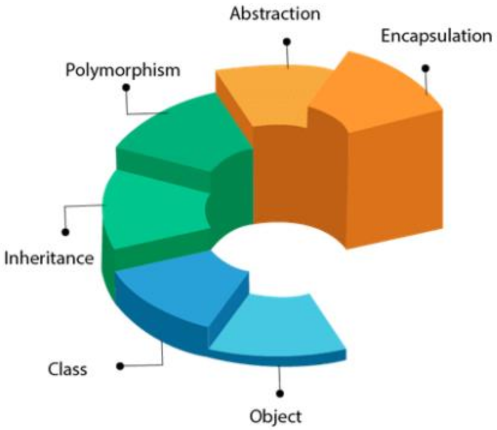
PROJECT TRAINING WORKSHOP

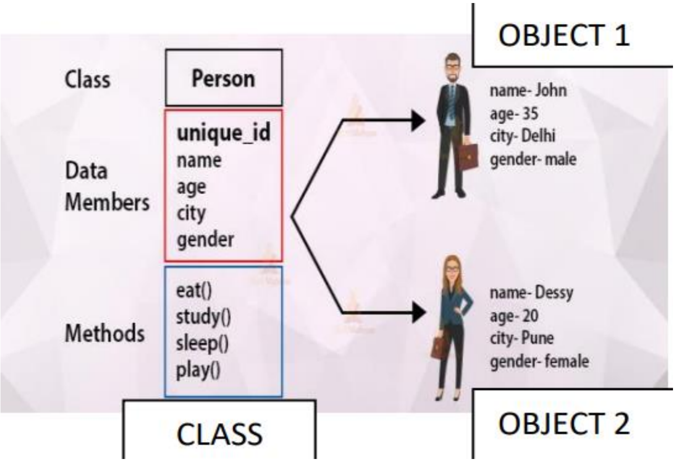
TECHNICAL OOPS

UNDERSTANDING OOPS

- **What is Object Oriented Programming (OOPS)?**
  - **The object-oriented programming is a different approach to programming. It has been created with a view to increase programmer's productivity by overcoming the weaknesses found in procedural programming approach.**







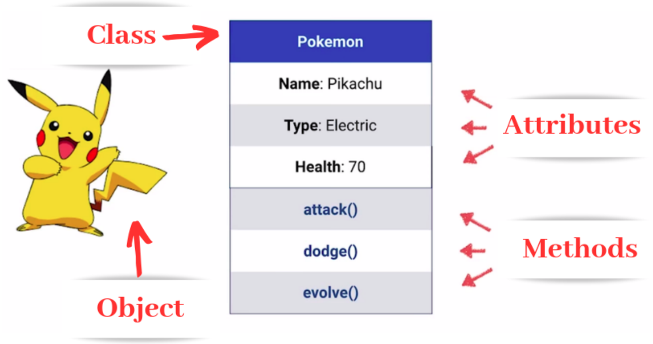
UNDERSTANDING OOPS

- **What are features/concepts of OOPS?**

Object	Class	Data Abstraction
Data Encapsulation	Modularity	Inheritance
Polymorphism		

- **What is Object?**
  - **Any real world thing that has state and behavior is known as object.**
  - **Also defined as an instance of a class.**

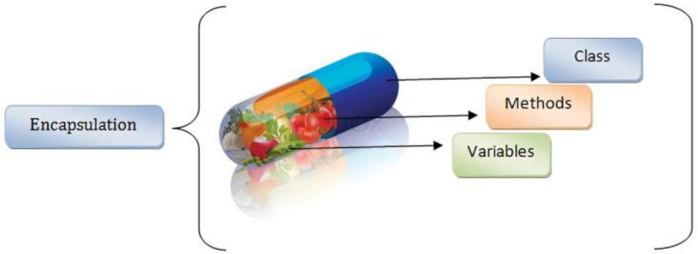
- **What is Class?**
  - **A class is a logical entity**
  - **Also defined as blueprint from which we can create individual object.**



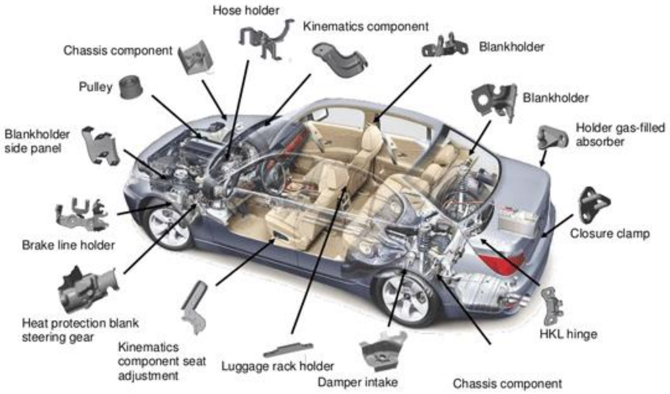
UNDERSTANDING OOPS

- **What is Data Abstraction?**
  - **Abstraction refers to the act of representing essential features without including the background details.**

- **What is Data Encapsulation?**
  - **Wrapping up of data and functions into a single unit is called as data encapsulation.**



- **What is Modularity?**
  - **The act of partitioning a program into individual components is called modularity.**

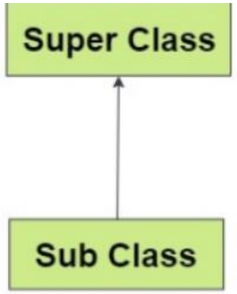


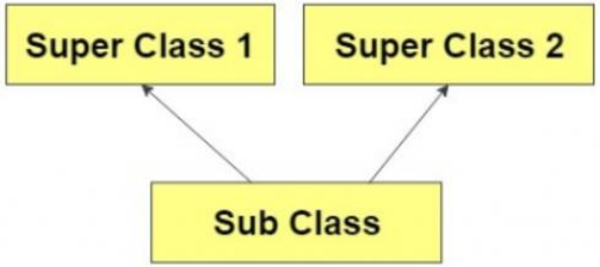
UNDERSTANDING OOPS

- **What is Inheritance?**
  - **It is a process by which object of one class inherit the properties of objects of another class.**

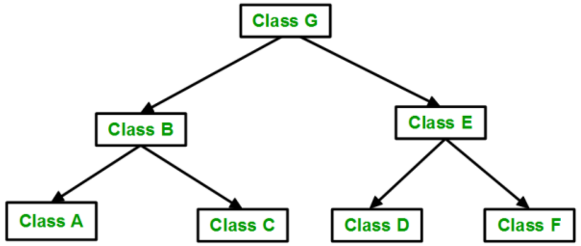
- **What are different types of Inheritance?**
  - **Single**
  - **Multiple**
  - **Hierarchical**
  - **Multilevel**
  - **Hybrid**

Single

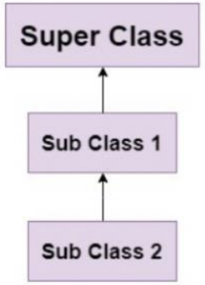




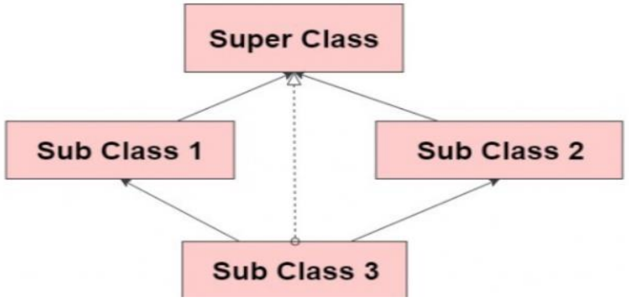
Multiple



Hierarchical



Multilevel

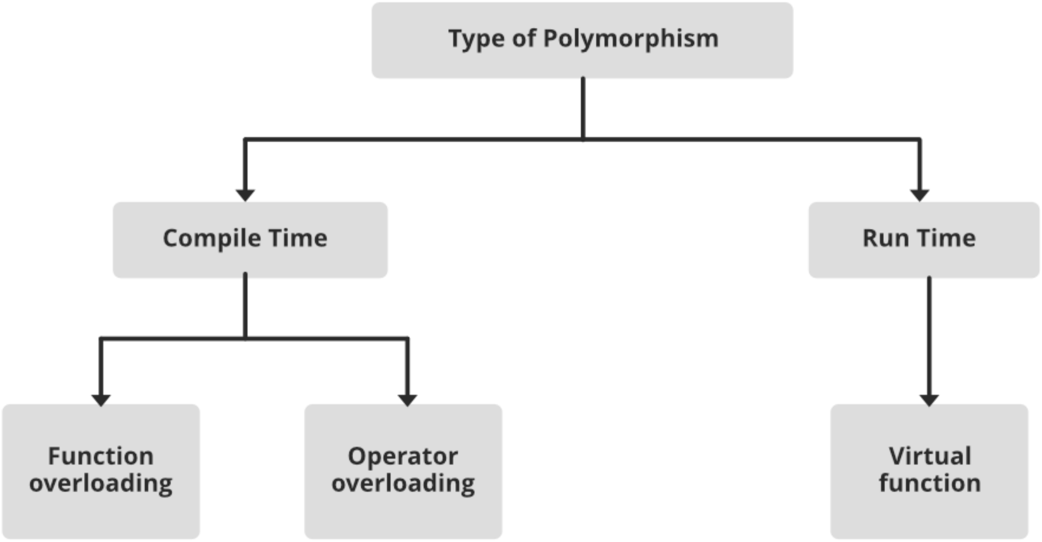


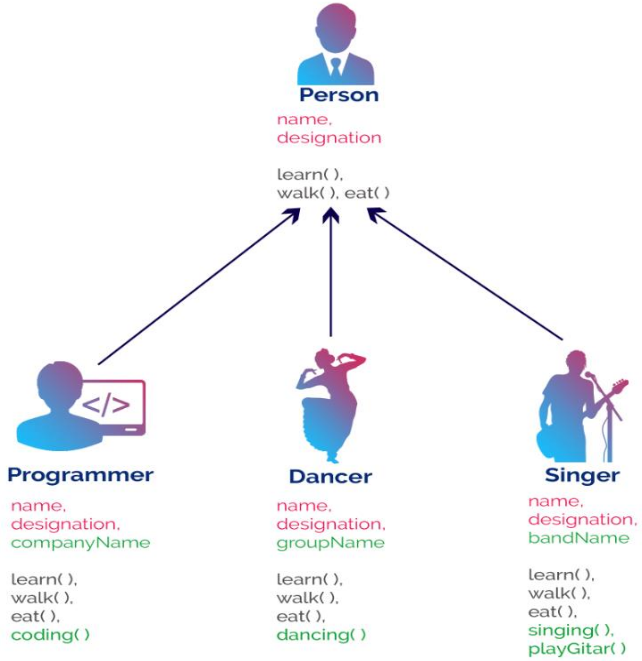
Hybrid

UNDERSTANDING OOPS

- **What is Polymorphism?**
  - **It is the concept that supports the capability of data to be processed in more than one form.**

- **What are different types of Polymorphism?**





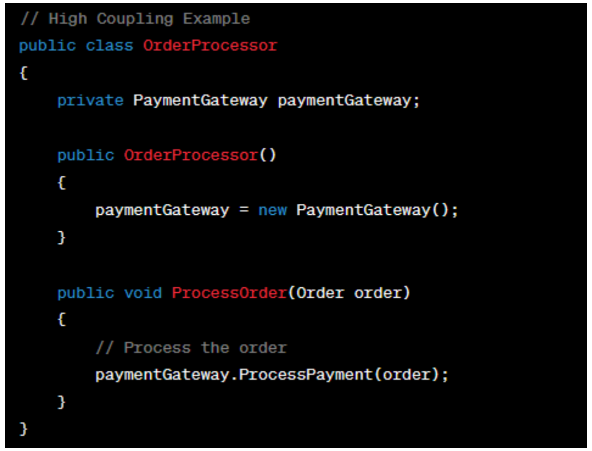
RAAFI
Builds You Best

UNDERSTANDING OOPS

- **What are other terms used in OOPS?**

Coupling	Cohesion	Association
Aggregation	Composition	

- **What is Coupling?**
  - **A measure of how much a module (package, class, method) relies on other modules.**
- **What are different types of Coupling?**
  - **Loose Coupling : Objects are independent to each other, does not modify the state of other**
  - **objects [GOOD HABIT]**
  - **Tight Coupling : Objects are dependent to each other, can modify the state**



```
// High Coupling Example
public class OrderProcessor
{
    private PaymentGateway paymentGateway;

    public OrderProcessor()
    {
        paymentGateway = new PaymentGateway();
    }

    public void ProcessOrder(Order order)
    {
        // Process the order
        paymentGateway.ProcessPayment(order);
    }
}
```

- **What is Cohesion?**
  - **A measure of how closely related the members (classes, methods, functionality within a method) of a module are to the other members of the same module.**
- **What are different types of Cohesion?**
  - **High Cohesion : Class that does well defined job. [GOOD HABIT]**
  - **Low Cohesion : Class that does lot of jobs.**

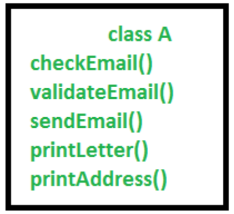


Fig: Low cohesion

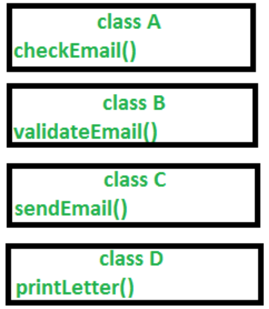
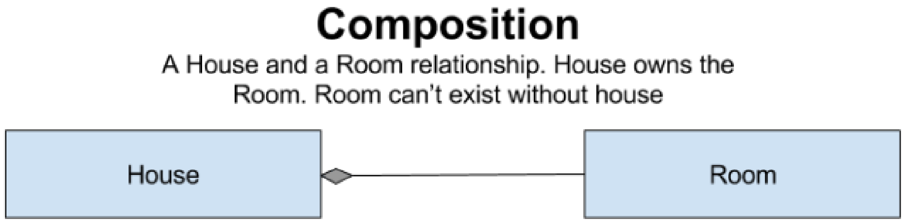


Fig: High cohesion

UNDERSTANDING OOPS

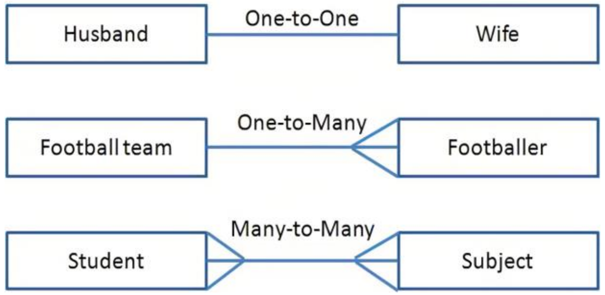
- **What is Composition?**
  - **A class that references one or more objects of other classes in instance variables. It allows us to model a has-a association between objects.**



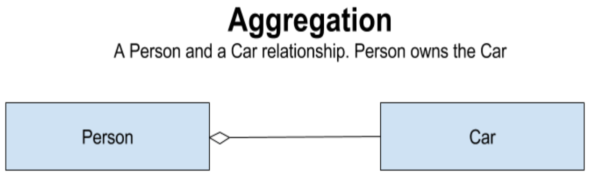
Composition

A House and a Room relationship. House owns the Room. Room can't exist without house

- **What is Association and its Types ?**
  - **Defines relationships between the objects**
  - **Types**
    - **One to One**
    - **One to Many**
    - **Many to One**
    - **Many to Many**

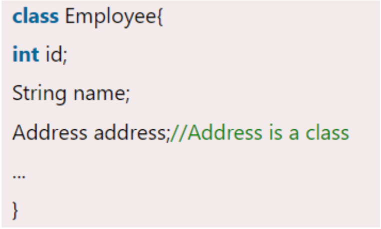


- **What is Aggregation?**
  - **It is a specialized form of Association where all object has its own lifecycle but there is ownership.**



Aggregation

A Person and a Car relationship. Person owns the Car



```
class Employee{
int id;
String name;
Address address;//Address is a class
...
}
```

TECHNICAL OOPS

- **ASSESSMENT**
  - **Take any real world object and apply the OOPS concepts with its operations**