OOPS Concept

Object: = State and Behavior

Car: - Blue, Brand

Behavior—Accelerate, Run Etc.

Procedural Programming: -

Whole Programming will be distributed into Binary

But every Thing is not able to solve the Programming.

# Real-world: -

Main Important Thing in the real World Is relationship.

How To deal with relationship

So, in the world of Programming for solving the relationship we Started to solve the Object oriented programming.

Concept of the Oops:

* Abstraction
* Encapsulation
* Inheritance
* Polymorphism.

Class And Object: -

Class is the Template With use this we made the object. And the Object is the Instance of the class.

Class Is the Template and using the template we make entity is called Object.

Suppose we take the shape of one circle which Are templeate for the bread that’s is the Class and If I make the bread with using .

If the Prograaming give u the Data Type than its primitive data Type

And if we create the data Type is called User Defined Data Types.

Abstraction: -

It is derived From the Abstract. It means Summary.

Small Important Things which I want to show. Show the only thing What I want to show to customer hide unimportant thing.

Real world Example: -

Like I m driving the Car, So I need to only know the Sterring, if I want to move left then we have to Mode the steering to left but we do not know How the wheel is Moving, How the Friction is reducing, How the Engine Is Working, all these complexities are Hided this is Called Abstraction.

Advantage of Abstraction: -

Loosely Coupled Application,

Make Easy to Use for User.

Security.

In java We implement Abstraction By using abstract Class and Interface.

So in Abstract Class We define the method And we explain About method in some other class by inheriting the Parent Class.

In Interface we Have to Write 100% so we can Tell The Interface IS 100% abstraction.

# Encapsulation

In real world entity :-

If I have a car which name is Anand , So I will not allow Anyone to Change the Name..

Same thing in programming I will not allow Anyone to change the state of object.

Real world

There is Name who have Car

Programming World

There is class, which have some state like name, Car car

We use the keyword name is Private before the variable so no one will be allowed to change the state directly.

So, in programming language we give the name is Access Modifier.

IF I want to create One channel so that some who want to change the State can come through channel and can change.

So I will use access modifier, That Keyword is Protected.

Without Encapsulation

If there is a house, if one my best friend come to house, they come any time and coming into kitchen and taking sugar, spoiling everything . So we can relate this with programming language .

.

If we add one Channel through the use come and use the Object.

So that channel is Getter and setter.

What we do we give access Modifier.

We give the private keyword that means we can use Within the class We create one method public which can be used through other object . That Method Is getter and setter.

Getter is only to read and Setter is to set or change the state.

**Constructor: -**

**Constructor is just like Method, but Its use to create instance of class ,I mean when u create the template but without creating instance .So creating instance mean that just giving memory in the Main Memory.**

**So constructor is same as method Which is used to create instance.**

**In Real world the relationship is checked based on DNA**

**But in the programming, DNA test is IS-A test.**

**If two classes, we want to check any relationship than we simply check Is-A relationship.**

**If the class IS-A relationship than only can be extended. Inheritance is Properties of OOPS.**

**Relationship: - Human Posses the Car.**

**In real World Possein we check the posseisn by Instagram,status etc.**

**But in Coding world the possein Relation is tested by HAS-A car. This type of relationship is called Association.**

**Class car{**

**String name;**

**String color;**

**}**

**Class Person{**

**String name;**

**Car car;**

**}// This is called Association.**

**There are two type of Association**

1. **Composition**
2. **Aggregation.**

**If university has Professor, if the professor died still university will run , and vice versa so that not totally dependent each other is Called Aggregation.**

**Human And heart-**

**If there is Heart than There will be human and If there is Human than There is Heart.**

**This is called Compostion.**

**Relationship**

**1.Parent Child- IS -A relationship.**

**This is achieved with extend or implement.**

**2.Posseeion relationship**

**. has A**

**1.Aggreagtion weak relationship**

**2. Composition Strong relationship**

**Polymorphism: -**

**Poly + Morphism: - Many + Form**

**Polymorphism Is of two Type :-**

**Static/ Compile Time polymorphism.**

**Dynamic /Run Time Polymorphism.**

**Static /Compile Time Polymorphism: -**

1. **Method Overloading: -**

**Ex: - add (int a, int b)**

**{**

**Return a+b;**

**}**

**Add (int a,int b,int c)**

**{**

**Return a+b+c;**

**}**

**This is called static and Compile time Polymorphism.**

**# Method OverLoading**

**Dynamic / Run Time Binding Polymorphism.**

**Method Overriding:-**

**For getting this we use Inheritance:-**

**Animal:- cat, Dog**

**One I create parent class Animal nd we extend that Class with subclass**

**Cat and Dog.**

**Class cat extends Animal**

**{**

**Eat()**

**{**

**Pos(“cat is eating”)**

**}**

**}**

**{**

**.**

**1.Inheritance: - We use parent child relationship., it help to reuse the code and reduce the code.**