

Session-1-In-Class_Python_OOP

OOP-Kavramlar

Training Clarusway

Pear Deck - April 8, 2022 at 5:36PM

Part 1 - Summary

Use this space to summarize your thoughts on the lesson

Part 2 - Responses

Slide 1



CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 2

Table of Contents ➤

- ▶ Fundamentals of OOP
- ▶ Objects
- ▶ Classes
- ▶ Recap

CLARUSWAY®
WAY TO REINVENT YOURSELF



Use this space to take notes:

Slide 3

Your Response

Did you understand OOP ?

Students, drag the icon!

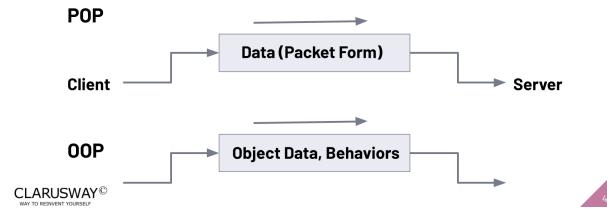
Pear Deck
Do not remove this box

Use this space to take notes:

Slide 4

►Object Oriented&Procedural Oriented»

- ▶ Object Oriented Programming (**OOP**) has become popular over recent years and has completely replaced the Procedural Oriented Programming (**POP**).



Use this space to take notes:

Slide 5



1

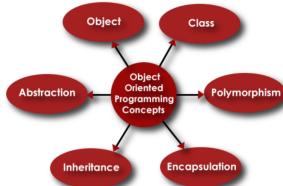
Fundamentals of OOP

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 6

▶ Fundamentals



CLARUSWAY®
your to success!



Use this space to take notes:

Slide 7

▶ Fundamentals



- ▶ **Object-Oriented Programming (OOP)** is a programming paradigm based on the concept of **objects** that interact with each other to perform the program functions.



CLARUSWAY®
your to success!



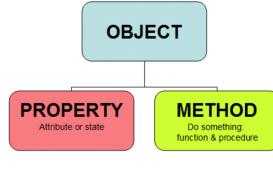
Use this space to take notes:

Slide 8

▶ Fundamentals



- ▶ Each object can be characterized by a **state** and **behavior**. An object keeps the current state in the **fields** and the **behavior** in the methods.



CLARUSWAY®
use to enhance learning

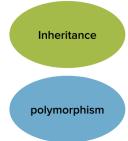
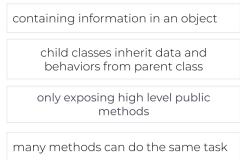
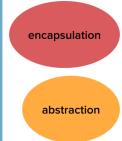
8

Use this space to take notes:

Slide 9

Your Response

Draw lines to match the image to the answer:
How is your pre-class level?



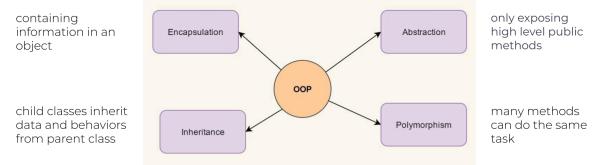
Students, draw anywhere on this slide!

Pear Deck Interactive Slide
Do not remove this bar

Use this space to take notes:

Slide 10

► Fundamentals



CLARUSWAY®
way to reinvent yourself

10

Use this space to take notes:

Slide 11



2 ► Objects

CLARUSWAY®
way to reinvent yourself

Use this space to take notes:

Slide 12

► Objects



- The key notion of the OOP is, naturally, an **object**.



12

Use this space to take notes:

Slide 13



3 ► Classes

CLARUSWAY®
WAY TO REINVENT YOURSELF

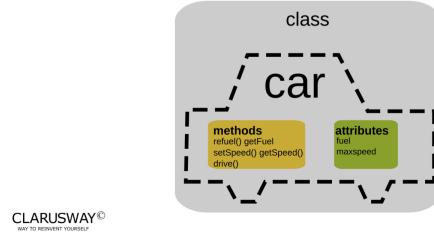
Use this space to take notes:

Slide 14

► Classes



- A **class** or **type** is a blueprint for the structure of methods and attributes.



14

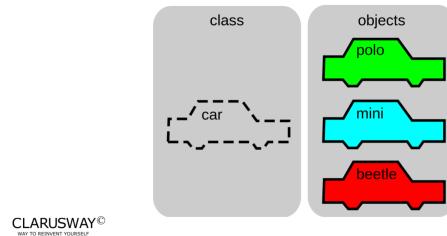
Use this space to take notes:

Slide 15

► Instances



- An object is an individual **instance** of a class.

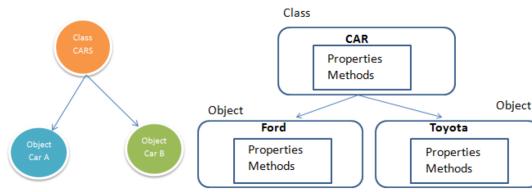


15

Use this space to take notes:

Slide 16

▶ Instance Sample



Car Ford = new Car(); Car Toyota = new Car();

CLARUSWAY®
your to innovative leadership

18

Use this space to take notes:

Slide 17

▶ Coding Sample

```
public class Car{
    private string _color;
    private string _model;
    private string _makeYear;
    private string _fuelType;

    public void Start(){
        ...
    }

    public void Stop(){
        ...
    }

    public void Accelerate(){
        ...
    }
}
```



CLARUSWAY®
your to innovative leadership

17

Use this space to take notes:

Slide 18

► Constructor



- **Constructor** is a special type of method called to create an object.

```
public class Demo {  
    Demo(){  
        ...  
    }  
    Demo(String s){  
        ...  
    }  
    Demo(int i){  
        ...  
    }  
    ...  
}
```

CLARUSWAY®
your to success

18

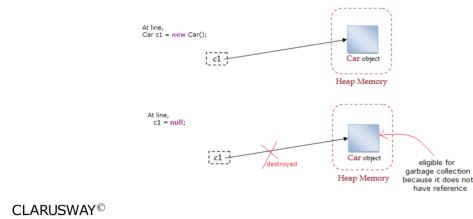
Use this space to take notes:

Slide 19

► Destructor



- **Destructor** is a member function which destructs or deletes an object.



CLARUSWAY®
your to success

19

Use this space to take notes:

Slide 20



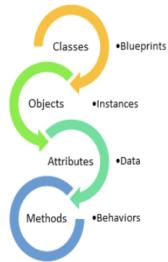
4 Recap

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 21

▶ Recap



CLARUSWAY®
WAY TO REINVENT YOURSELF

21

Use this space to take notes:

Slide 22

Basic Principles of OOP



CLARUSWAY®
WAY TO REINVENT YOURSELF.

Use this space to take notes:

Slide 23

Table of Contents

- ▶ Encapsulation
- ▶ Abstraction
- ▶ Inheritance
- ▶ Polymorphism

CLARUSWAY®
WAY TO REINVENT YOURSELF.

Use this space to take notes:

Slide 24



1 Encapsulation

CLARUSWAY®
WAY TO REINVENT YOURSELF

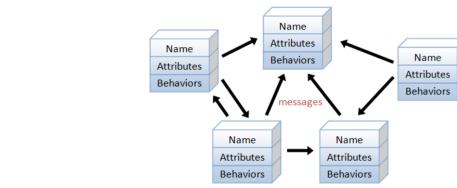
Use this space to take notes:

Slide 25

▶ Encapsulation



- An object-oriented program consists of many well-encapsulated objects and interacting with each other by sending **messages**.



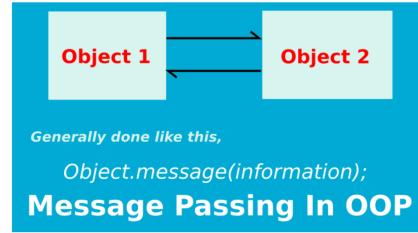
CLARUSWAY®
WAY TO REINVENT YOURSELF

25

Use this space to take notes:

Slide 26

► Message Sending



CLARUSWAY®
your to innovative technology

28

Use this space to take notes:

Slide 27

► Domain-Specific Methods

- Operations that how to control the **changes of the data** through methods.

Object 1		Object 2	
Model	Volkswagen Polo	Model	Volkswagen Vento
Fuel	Petrol	Fuel	Diesel
Make	2017	Make	2017
Start()		Start()	
Break()		Break()	
Accelerate()		Accelerate()	

CLARUSWAY®
your to innovative technology

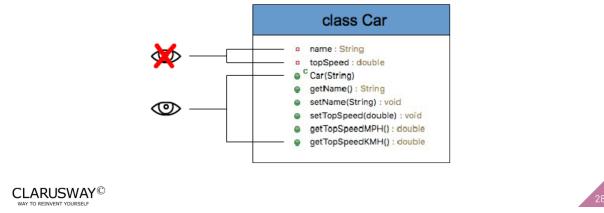
27

Use this space to take notes:

Slide 28

► Getters and Setters

- ▶ **Setter** and **Getter** methods to modify and view the variables values.

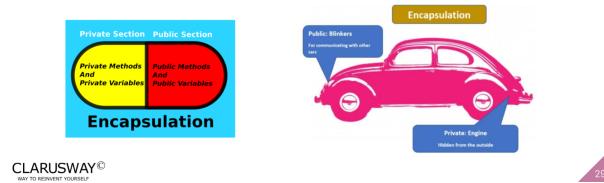


Use this space to take notes:

Slide 29

► Protecting Data

- ▶ **Encapsulation** adds **security** to code and makes it easier to **collaborate** with external developers. Within classes, most programming languages have **public**, **protected** and **private** sections.



Use this space to take notes:

Slide 30



2 ➤ Abstraction

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 31

➤ Abstraction



- ▶ A process of **hiding** the implementation details from the user.



CLARUSWAY®
WAY TO REINVENT YOURSELF

31

Use this space to take notes:

Slide 32



3 Inheritance

CLARUSWAY®
WAY TO REINVENT YOURSELF

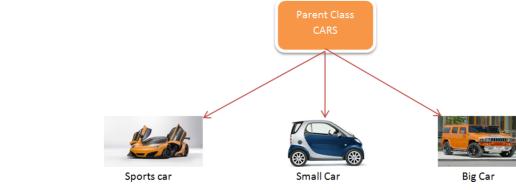
Use this space to take notes:

Slide 33

Inheritance



- Inheritance allows **child classes** to inherit features of **parent classes**.



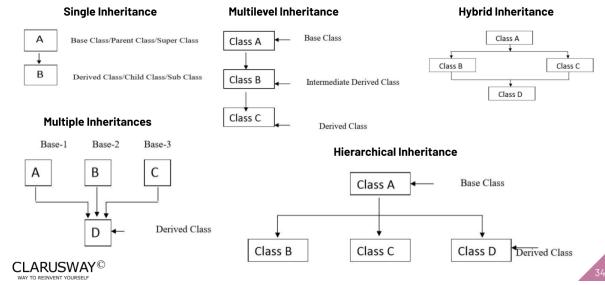
CLARUSWAY®
WAY TO REINVENT YOURSELF

33

Use this space to take notes:

Slide 34

► Types of Inheritance

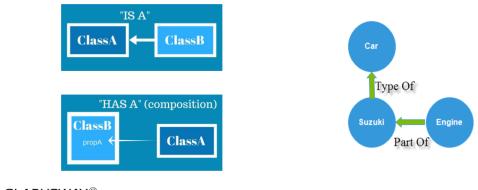


Use this space to take notes:

Slide 35

► Types Of Relationship

- One of the advantages of Object-Oriented programming language is code **reuse**. This reusability is possible due to the **relationship** b/w the classes.



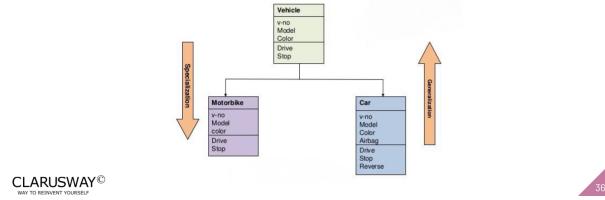
35

Use this space to take notes:

Slide 36

► Generalization / Specialization

- ▶ Vehicle generalize what is common between Car and Motorbike.
- ▶ Car and Motorbike specialize Vehicle to their own sub-type.

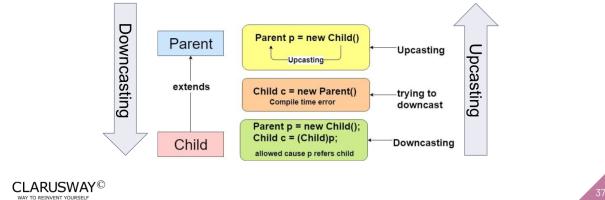


Use this space to take notes:

Slide 37

► Type Casting

- ▶ Upcasting is casting a subtype to a supertype.
- ▶ When Subclass type refers to the object of Parent class, it is known as downcasting.



Use this space to take notes:

Slide 38



4 Polymorphism

CLARUSWAY®
WAY TO REINVENT YOURSELF

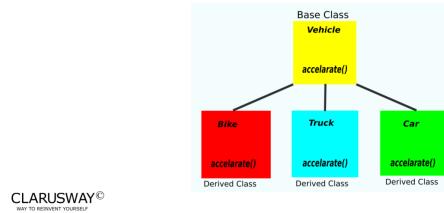
Use this space to take notes:

Slide 39

▶ Polymorphism



- ▶ Poly means "many" and morphism means "form". Polymorphism means objects can take **different forms** under different conditions.



CLARUSWAY®
WAY TO REINVENT YOURSELF

39

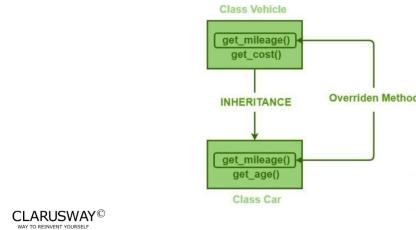
Use this space to take notes:

Slide 40

► Overriding



- ▶ A child class can provide a **different implementation** than its parent class.



CLARUSWAY®
your to success!

40

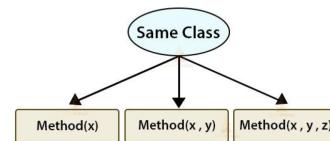
Use this space to take notes:

Slide 41

► Overloading



- ▶ Methods or functions may have the same name, but a **different number of parameters** passed into the method call.



CLARUSWAY®
your to success!

41

Use this space to take notes:

Slide 42

▶ Introduction to Interfaces



CLARUSWAY®
WAY TO REINVENT YOURSELF.

Use this space to take notes:

Slide 43

Table of Contents

- ▶ Basic Definitions
- ▶ One-Method Interface
- ▶ Complex Interface
- ▶ Responsibility of an Interface

CLARUSWAY®
WAY TO REINVENT YOURSELF.

Use this space to take notes:

Slide 44



1 ▶ Basic Definitions

CLARUSWAY®
WAY TO REINVENT YOURSELF

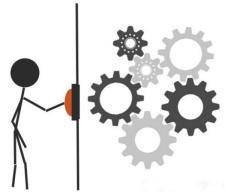
Use this space to take notes:

Slide 45

▶ Basic Definitions



- ▶ An **interface** is a collection of methods that describes the behavior of an object.



CLARUSWAY®
WAY TO REINVENT YOURSELF

45

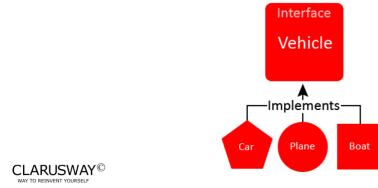
Use this space to take notes:

Slide 46

► Basic Definitions



- ▶ A class that **implements** an interface must implement all the methods declared in the interface. The methods must have the exact same **signature** (name + parameters) as declared in the interface.



48

Use this space to take notes:

Slide 47



2 ► One-Method Interface

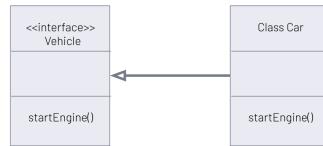
CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 48

► One-Method Interface

- One method stands for only **one skill** of an object.



CLARUSWAY®
WAY TO REINVENT YOURSELF

48

Use this space to take notes:

Slide 49



3 ► Complex Interface

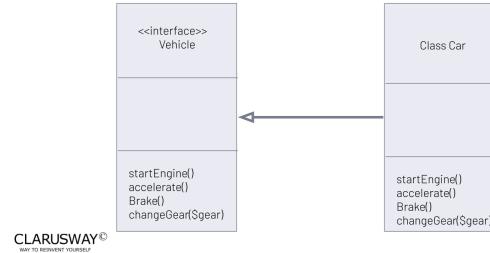
CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 50

► Complex Interface

- ▶ Many methods stand for **many skills** of an object.



50

Use this space to take notes:

Slide 51



4 ► Responsibility of an Interface

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 52

► Responsibility of an Interface



- ▶ Defining a system's **boundaries**.
- ▶ Depicting the **dependencies** of a system.
- ▶ **Coordinating** with various parties.
- ▶ Ensuring **compatibility** among systems.
- ▶ Exposing potential **problem areas** and **risks**.

CLARUSWAY®
way to reinvent yourself

52

Use this space to take notes:

Slide 53



► Recap

CLARUSWAY®
way to reinvent yourself

Use this space to take notes:

Slide 54

► Recap



Use this space to take notes:

Slide 55

THANKS!

Any questions?

CLARUSWAY®
way to reinvent yourself



Use this space to take notes: