Object Oriented Programming using Python (I)

Lecture(6)

Object Oriented Programming With Real-World Scenario

Prepared by: ***Dr. Rula Amjed Hamid***

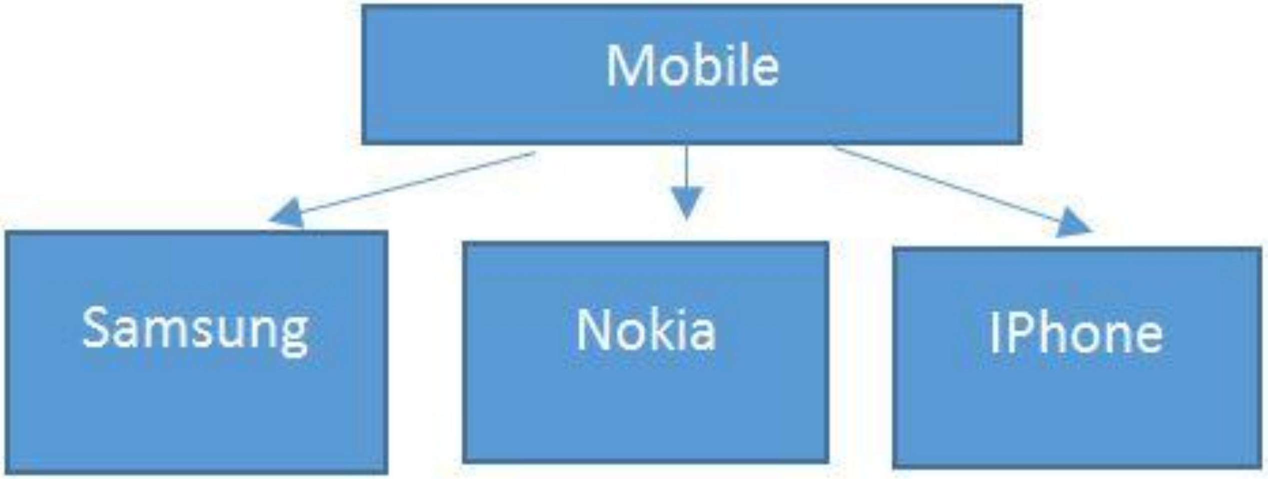
University of Information Technology and Communications

College of Business Informatics

Object Oriented Programming With Real-World Scenario

Let’s consider an example for explaining each concepts which at the end will make you understand & follow Object Oriented Programming.

When we take a mobile as an object, its basic functionality for which it was invented were Calling & Receiving a call & Messaging. But now a days thousands of new features & models were added & the count is still increasing.



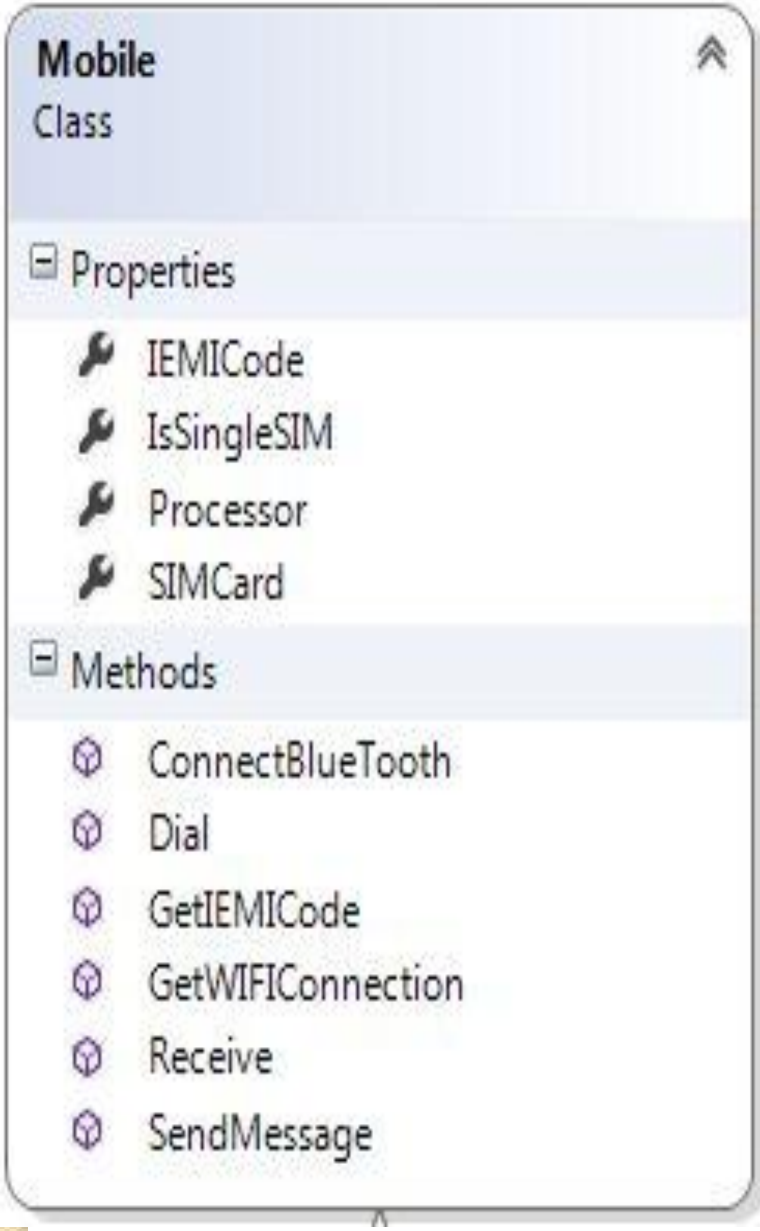
In above diagram, each brand (Samsung, Nokia, IPhone) have their own list of features along with basic functionality of dialing, receiving a call & messaging.

Objects

-Any real world entity which can have some characteristics or which can perform some work is called as Object.
-This object is also called as an instance i.e. - a copy of entity in programming language.
-If we consider the above example, a mobile manufacturing company, at a time manufactures laces of pieces of each model which are actually an instance. This objects are differentiated from each other via some identity or its characteristics. This characteristics is given some unique name.

Class

-A Class is a template which describes the object. Mainly a class would consist of a name, attributes & operations.
-Considering the above example, A Mobile can be a class which has some attributes like Profile Type, IMEI Number, Processor, and some more.) & operations like Dial, Receive & SendMessage.



Mobile
Class

Properties

- IEMICode
- IsSingleSIM
- Processor
- SIMCard

Methods

- ConnectBlueTooth
- Dial
- GetIEMICode
- GetWIFIConnection
- Receive
- SendMessage

Mobile Tools	Function
SMS	To send text messages
Camera	To take a photo
MMS	To send photo and video
Recorder	For recording sound/video
MP3/MP4	For listing what was already recorded
Memory	For data storage
Apps	For different purposes (e.g.communication)
Browsers	To navigate the Internet
GPS	To provides geolocation and time information
QR Code	To scan and generate for accessing information

Abstraction

-Abstraction says, only show relevant details and rest all hide it. This is most important in OOPS as it is providing us the technique to hide irrelevant details from User.
-If we consider an example of any mobile like Nokia, Samsung, IPhone.

Some features of mobiles

- Dialing a number call some method internally which concatenate the numbers and displays it on screen but what is it doing we don't know.

- Clicking on green button actual send signals to calling person's mobile but we are unaware of how it is doing.

-This is called abstraction where creating method which is taking some parameter & returning some result after some logic execution without understating what is written within the method

Encapsulation :

-is defined as the process of enclosing one or more details from outside world through access right.

-It says how much access should be given to particular details.

-Both Abstraction & Encapsulation works hand in hand because Abstraction says what details to be made visible & Encapsulation provides the level of access right to that visible details. i.e. – It implements the desired level of abstraction.

Talking about Bluetooth which we usually have it in our mobile. When we switch on the Bluetooth I am able to connect another mobile but not able to access the other mobile features like dialing a number, accessing inbox etc. This is because, Bluetooth feature is given some level of abstraction.

-Another point is when mobile A is connected with mobile B via Bluetooth whereas mobile B is already connected to mobile C then A is not allowed to connect C via B. This is because of accessibility restriction.
This is handled by access specified like public, private, protected, and internal

Polymorphism

Polymorphism can be defined as the ability of doing the same operation but with different type of input.

Let’s say Samsung mobile have the 5MP camera available i.e. – it is having a functionality of CameraClick(). Now same mobile is having Panorama mode available in camera, so functionality would be same but with mode. This type is said to be polymorphism

Inheritance

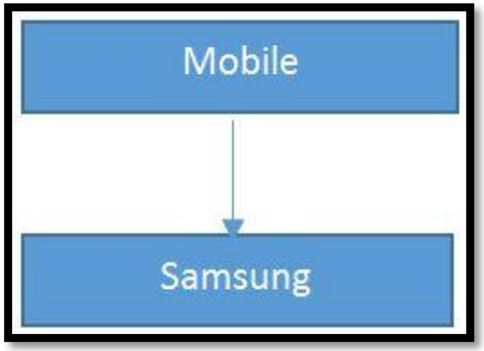
Ability to extend the functionality from base entity in new entity belonging to same group.
This will help us to reuse the functionality which is defined before .

Considering the example, Basic Mobile functionality is to Send Message, dial & receive call. So the brands of mobile is using this basic functionality by extending the mobile class functionality and adding their own new features to their respective brand.

There are mainly 4 types of inheritance:

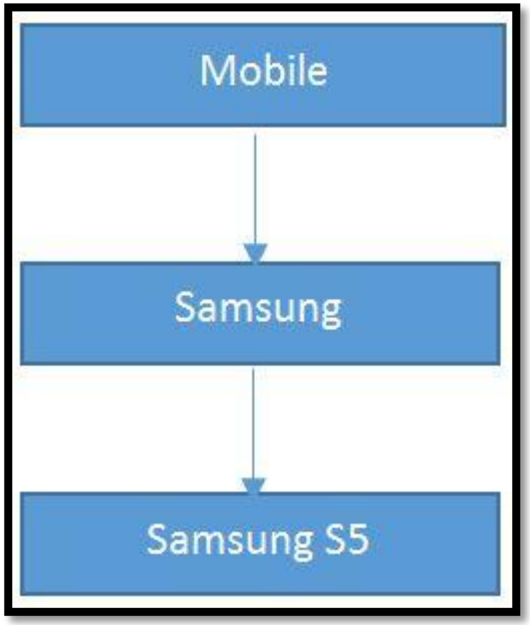
- Single level inheritance
- Multi-level inheritance
- Hierarchical inheritance
- Hybrid inheritance

Single level inheritance



In Single level inheritance, there is single base class & a single derived class i.e. - A base mobile features is extended by Samsung brand.

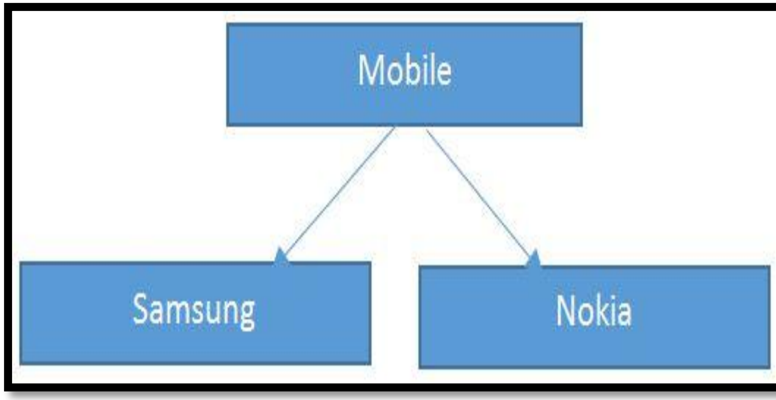
Multilevel inheritance



In Multilevel inheritance, there is more than one single level of derivation. i.e. - After base features are extended by Samsung brand. Now Samsung brand has manufactured its new model with new added features or advanced OS like Android OS, v4.4.2 (kitkat). From generalization, getting into more specification.

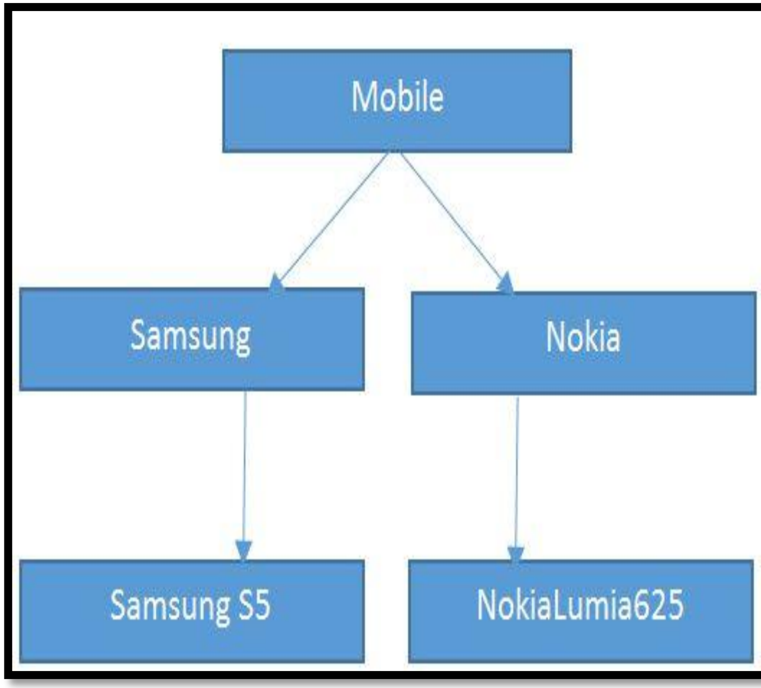
Hierarchal inheritance

In this type of inheritance, multiple derived class would be extended from base class, it's similar to single level inheritance but this time along with Samsung, Nokia is also taking part in inheritance



Hybrid inheritance

Single, Multilevel, & hierarchal inheritance all together construct a hybrid inheritance.



Summary:

Class:A category of objects. The class defines all the common properties of the different objects that belong to it.

Object: a self-contained entity that consists of both data and procedures to manipulate the data.

Abstraction: The process of picking out (abstracting) common features of objects and procedures.

Information hiding: The process of hiding details of an object or function. Information hiding is a powerful programming technique because it reduces complexity.

Encapsulation: The process of combining elements to create a new entity. A procedure is a type of encapsulation because it combines a series of computer instructions.

Inheritance: a feature that represents the "is a" relationship between different classes.

Polymorphism: A programming language's ability to process objects differently depending on their data type or class.

Interface: the languages and codes that the applications use to communicate with each other and with the hardware.

Messaging: Message passing is a form of communication used in parallel programming and object-oriented programming.

Procedure: a section of a program that performs a specific task.