**OOP Fundamentals:**

**Class:**

* Code template for creating program objects as many as we want.
* Class components: 1) Name (a.k.a type) 2) Attributes (a.k.a properties, data) 3) Behaviour (a.k.a methods, operations)
* Method: A program procedure that can return a value. It is defined as a part of class. They can only access data known to it’s object
* After defining the class, creating objects = instantiation (Each object created is an instance of that particular class)

**Principles of OOP paradigm: (AEIP)**

(i) Abstraction:

It refers to the concept of hiding the complexities (i.e. unnecessary details) of a system from the users of that system.

(ii) Encapulation:

It refers to the concept of bundling of data with the methods that operate on that data or the restricting of direct access to some of an object’s components.

(iii) Inheritance:

It bases a new object or class on an existing one and lets you inherit the existing attributes and methods.

(iv) Polymorphism:

It is the ability of an object to take on many forms.

5-step process of Object Oriented Design:

1) Gather requirements

2) Describe the application

3) Idenitfy the main objects

4) Describe the interactions

5) Create a class diagram

Unified Markup Language (UML): Standardized notation for diagrams to visualize Object Oriented systems.

FURPS Requirements for Software Design:

* Functionality : Capability. Reusability, Security
* Usability : Human factors, Aesthetics, Consistency, Documentation
* Reliability : Availability, Failure Rate & Duration, Predictability
* Performance : Speed, Efficiency, Resource consumption, Scalability
* Supportability : Testability, Extensibility, Serviceability, Configurability