OOPS

- Objects - Real World Entity
    1. Properties or State or Data variables
    2. Behaviours (Functionality) or Functions or Data methods
- procedural programming - importance to function, not data. (Data can move freely)
- oops programming - data hiding (binds function and data)

1. **Data Abstraction**
    - Can you explain OOPS to 14 year old child ?
        - Don't use any jargons
    - Hides internal implementation and shows only essential functionality to user.
    - Example - Cell phone, when dial number and press green button, call will be made. (Internal working we don't know)
    - Interface and abstract classes
    - Helps in security and confidentiality
    -
2. **Data Encapsulation**
    - Data hiding
    - Bundles data, and data functions into a single unit
    - Better control
    -
3. **Inheritance**
    - Child class inheriting properties, methods from parent class
    - Opposite is not true
    - There are different types of inheritance
        - Single, Hierarchical, Multi-level allowed
        - Multiple not allowed - because there might be similar name = properties, or methods on parent classes
        -
4. **Polymorphism**
    - Many forms of methods
    - Same methods performs differently, in different situations.
    - Method Overloading / Static Polymorphism / Compile Time polymorphism
        - Same method name with different parameters, (type, numbers)
        - Known during compile time
        - On basis of return type, overloading cannot be done (can not determine which function to invoke)
    - Method Overriding / Runtime polymorphism / Dynamic polymorphism
        - Method defined again in child class with same arguments, with same return type
        - Only functionality changes

- Is-a Relationship

    - Achieved through inheritance
- Has -a Relationship
    - Inside particular class, we have object of another class.
    - Weak - Aggregation
    - Strong - Composition