OOPS Java Concepts:

1) **Method Programming:**
   - ✓ AM Return type Method name{
   }
   - Using Object:
     Variable declare in class/dis() method/main method/obj for class/Assigning value to Variable/call Method()
   - Using Method:
     - ➢ Without Parameter & without Return type
     - ➢ With Parameter & With Return type
     - ➢ Without Parameter & With Return Type
     - ➢ With Parameter & Without Return Type
2) **Constructor:**
   - ✓ Used for initializing instance variable (empid=20;)
   - ✓ Same name as class name
   - ✓ No Return type
   - ✓ Overloading possible

   - Default
   - Parameterized (this pointer use)
3) **Inheritance:**
   - ✓ Acquiring methods and variables from parent class to child class
   - ✓ Extends Keyword
     - Single-level -Java Support
     - Multi-Level -Java Support
     - Hierarchal -Java Support
     - Hybrid -Interface Support
     - Multiple -Interface Support
4) **Polymorphism:**
   - ✓ One thing in Multiple Forms
   - ✓ Same method but different implementations

- Method Overloading (Compile time/early Binding)
  - ✓ Single class but same name 'n' no of methods ()

    1. Overload by changing no of arguments
    2. Overload by changing datatype of arguments
    3. Overload by changing order of parameters

- Method Overriding (Runtime/Late binding)
  - ✓ parent class method child class Override
  - ✓ Declaring method in child class which is already defined in parent class
  - ✓ Super keyword invokes immediate parent class method/constructor/instance variable

    1. Right click→sources→override/implements methods→method name

5) **Abstraction:**
   - ✓ Hiding implementation details from user
   - ✓ 0-100% not complete hiding
   - ✓ Abstract keyword class Google {}
   - ✓ Cannot create abstract class objects
   - ✓ We will use Inheritance(extends)
   - ✓ Crete object for child class

     - Abstract class
     - Abstract Method (method with body)
     - Non-Abstract Method (method without body)

6) **Interface:**
   - ✓ Another way to achieve abstraction in java
   - ✓ Interface interface name {methods/variable declaration}
   - ✓ Final and static variable
   - ✓ Contains abstract methods
   - ✓ Implements Keyword
   - ✓ Cannot create objects
     - Hybrid
       - Combination of two or more inheritance

- Multiple
    - One child will 2 parent details

7) **Encapsulation:**
    - ✓ Complete Data hiding (100%)
    - ✓ Make private variables
    - ✓ Right click→source→generate getters setters
    - ✓ Getters and setters
8) **Exception:**
    - ✓ Unexpected condition which causes pgm to exit/terminate
        (i) Object (java.lang.object)→ Superclass
        (ii) Throwable class
        (iii) Exception class
        (iv) Compile time Exception (checked)
            - File not found
            - Interrupted Exception
            - I/O Exception

        (v) Runtime Exception (unchecked)
            - Arithmetic
            - Null pointer
            - Array index out of bound exception
9) **Exception Handling:**
    - Try → Exception statement
    - Catch →get message for exception
    - Finally, →mandatory executable statements
    - Throws→if we want to execute without handling