

Day 3

Classification of languages

1. Procedure oriented programming language
 - example: C programming language
 - It follows top-down approach
 2. Object oriented programming language
 - example: Simula, C++, Java etc, python, C#.
 - It follows bottom-up approach
 3. Object based programming language
 - example: Ada, Java Script, Visual Basic etc.
 4. Functional oriented programming language
 - example: Python, Scala, Java etc.
- Java is object oriented programming language.
 - Alan Kay is inventor of OOPS(Object oriented programming structure/system).
 - OOPS is not a syntax. It is a process/methodology.
 - Object oriented programming is a kind of programming in which, we can solve real world problems using class and object/instance.
 - Book : Object oriented analysis and design with applications
 - Autor : Grady Booch
 - According grady booch, there are 4 major and 3 minor pillars of oops.

4 Major pillars

1. Abstraction
2. Encapsulation
3. Modularity
4. Hierarchy

3 Minor pillars

1. Typing / Polymorphism
2. Concurrency
3. Persistence

Class

- If we want to group related data elements together then we should define class.
- class is a keyword in Java.

```
class ClassName{
    //TODO : declare member
}
```

- Class can contain:
 1. Nested type(s)(Interface, class, enum)
 2. Field(s)
 3. Constructor(s)
 4. Method(s)[Static / Non Static(final / abstract)]

How to solve real world problem using oops?

1. Read and understand problem statement.
2. Analyze problem statement and then design classes for it.
3. According requirement declare fields inside class.(Variable declare inside class is called field).
4. Instantiate class i.e create instance of a class.(Process of creating instance from a class is called instantiation). Instance will contain data/value/state.
5. If we want to process state of instance then we should call method on it.(Process of calling method on instance is called message passing).
6. Define method and usign this reference perform operations on object/instance.

Instance

- In Java, object is also called as instance.
- If we want to create instance of a class then we should use new operator.

```
Employee emp;           //Object reference / reference
new Employee( );        //Instance W/O reference / annonymous instance
```

- Instantiation

```
Employee emp;
emp = new Employee();    //OK
//or
Employee emp = new Employee( ); //OK
```

- Following members do not get space inside instance:
 1. Method parameter
 2. Method local variable
 3. Nested type
 4. Method(including constructot)
 5. Static field
- Only non static fields get space inside instance.
- Fields get space once per instance but according order of their declaration inside class.

Coding Convention

- Pascal Case Coding convention
 - In this case, including first word, first character of each word should be in uppercase.
 - Example:
 1. System
 2. StringBuffer
 3. NumberFormatException
 4. ArrayIndexOutOfBoundsException
 - We should use pascal case coding convention for:
 1. Type name(Interface, class, enum etc)
 2. File Name
- Camel Case coding convention
 - In this case, excluding first word, first character of each word should be in uppercase.
 - Example
 1. main
 2. parseInt
 3. showInputDialog
 4. addNumberOfDays
 - We should use pascal case coding convention for:
 1. Method Parameter
 2. Local Variable
 3. Field
 4. Method
 5. object reference
- Coding convention for package
 1. Package name should be in lowercase and it should be in following format.
 - java.lang
 - java.lang.reflect
 - com.mysql.cj.jdbc
 - javax.servlet.jsp
 - org.sunbeam.dac.test
- Name of final variable/field and enum constant should be in upper case.

Access Modifier

- If we want control visibility of members of class then we should use access modifier.
- Access modifiers in Java:
 1. private
 2. package/package level private / default
 3. protected
 4. public

```
class Test{  
    private int num1;  
    int num2;    //default-> package level private  
    protected int num3;
```

```
    public int num4;
}
```

- In java, default access modifier of members of class is package level private/default.

```
class Test{ //Concrete class
    private int num1;    //Non static Field / instance variable
    private static int num2;    //static field / class level variable
    //Concrete Method
    public void showRecord( ){ //Non static method / instance method
        //TODO
    }
    //Concrete Method
    public static void displayRecord( ){    //Static method / class level
method
        //TODO
    }
}
class Program{
    public static void main(String[] args) {
        Test t = new Test( );    //Instantiation
        t.showRecord( );
        Test.displayRecord( );
    }
}
```

Characteristics of instance

State

- Value stored inside instance is called state of instance.
- Value of the field represents state of the instance.

Behavior

- Set of operations that we can perform/call on instance represents its behavior
- Methods defined inside class represents behavior of the instance.

Identity

- A value of field that is used to identify instance uniquely is called identity of the instance.

Class

- class is collection of fields and methods.
- Structure and behavior of instance, depends on class hence class is considered as a template/model/blueprint for instance.
- Class is set of / collections of such objects which is having common structure and common behavior.
- Class is imaginary/logical entity.

- Example : Book, Mobile Phone, Car etc.
- Class implementation represents encapsulation.

Instance

- In Java, object is also called as instance.
- An entity, which has physical existence is called instance.
- An instance, which has state, behavior and identity is called instance.
- Instance is real time / physical entity.
- Example : Effective Java, One Plus 8, Tata nano etc.