**WHAT IS OOPS?**

OOPS, or Object-Oriented Programming System, is a way to write computer programs using "objects" that represent real-world things/Entities. These objects can store information (Data) and perform actions (Methods).

1. Object oriented programming (OOP) is a core of Java programming which is used for designing a program using classes and object.
2. The basic concept of OOPS is used to create objects and manipulate (modification) this object is to get an expected result.

WHAT ARE THE VARIOUS OOPS CONCEPTS IN JAVA?

1)Class

2) Object

3) Encapsulation

4) Polymorphism

5) Inheritance

6) Abstraction

7) Association

8) Composition

9) Interface

WHAT ARE THE PRINCIPLES OF OOPS?

1) Encapsulation

2) Inheritance

3) Polymorphism

4) Abstraction

NOTE: ***Concepts and Pillars are the different things. We have 9 main concepts in java and we have 4 pillars in java.***

The four fundamental principles of OOP —

1. encapsulation,
2. abstraction,
3. inheritance,
4. polymorphism
5. **What is Class?**

Class is user defined datatype.

Class is a template or a blueprint. which contains variables and functions.

1. **What is instance?**

dynamic memory allocation in the RAM for anything called as instance

1. **What is Object?**

dynamic memory allocation in the RAM for class called as Object.

An object is an instance of a class.

From one class we can create multiple objects.

**Creating Objects: Instances of Classes**

An object is an instance of a class. When a class is defined, no memory is allocated until an object of that class is created in the RAM(Heap).

**The object has its own state, behavior, and identity.**

It’s the implementation of the class blueprint.

Creating an object in Java involves two steps:

1. **Declaration:**A variable of the class type is declared.
2. **Instantiation:** Using the new keyword, memory is allocated for the object. Object Creation.

How we create object in Java. – using new keyword followed by classname

Employee emp; // **Declaration**

emp = new Employee(); // **Instantiation**

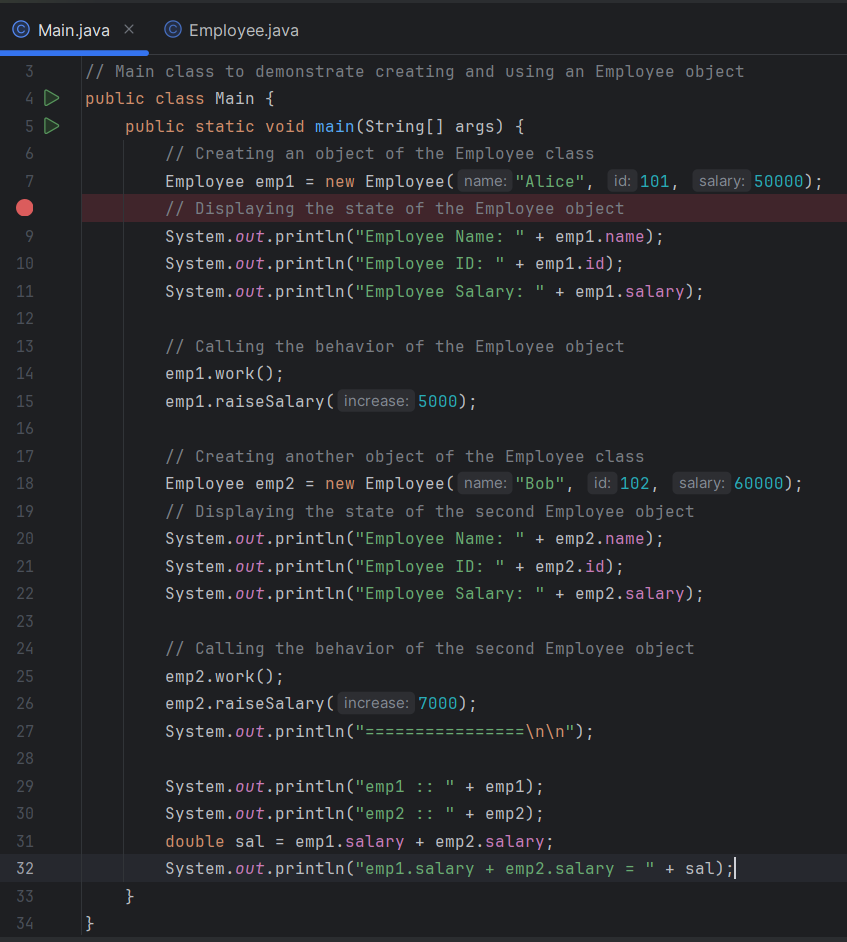
or

Employee emp = new Employee();

emp 🡪 Reference variable, object, instance

A screen shot of a computer program

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

A screenshot of a computer

Description automatically generated

**What is need of Object?**

To access members of class<**Person**> (variables & Functions) in the program which is under execution<Main>.

A screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

Main class is under execution because it has main() method. It want to access variable & Functions of person class, so object is created.

**Explanation:**

1. **Class Definition (Person):**
   * The Person class has two instance variables: name and age.
   * It also has a instance method displayInfo() that prints the person's name and age.
2. **Main Class (Main):**
   * In the main method, we create two objects of the Person class (person1 and person2).
   * For person1, we set the name to "John Doe" and age to 30, then call the displayInfo() method to display this information.
   * For person2, we set the name to "Jane Smith" and age to 25, then call the displayInfo() method to display this information.

By creating objects (person1 and person2), we can access and manipulate the members (variables and functions) of the Person class. This demonstrates the need for objects to interact with the class members during program execution.

Variables = properties =fields= attributes=state

What is state & behavior of an object?

* **State**: The state of an object is represented by its Variables or properties. These variables store information about the object.

-data present into object at that time of moment

* **Behavior**: The behavior of an object is represented by its methods (functions). These methods define what the object can do or how it can interact with other objects.

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

OPPS Reference: <https://medium.com/@AlexanderObregon/a-guide-to-object-oriented-programming-in-java-89dc4544837f#:~:text=The%20four%20fundamental%20principles%20of,hidden%20from%20the%20outside%20view>.

**Encapsulation**

Encapsulation is the process of bundling/wrapping related data (variables) and the code (methods) together into a single unit, called a class.

In encapsulation, the data in a class is hidden from other classes, so it is also known **as data-hiding.**

***We can achieve encapsulation in Java by:***

* Declaring the variables of a class as private.
* Providing public setter and getter methods to modify and view the variables values.

Need – to provide security to the data

To prevents the data from being accessed directly by external processes/programs/Classes, and instead, it is accessed through methods within the class.

**For example,** consider a class BankAccount, which encapsulates properties like accountNumber and balance. The only way to access and modify these properties is through methods like deposit or withdraw, ensuring that the account balance cannot be changed arbitrarily.

|  |  |
| --- | --- |
| A screenshot of a computer code  Description automatically generated | Explanation:  In this BankAccount class, the balance is encapsulated and protected from direct access. It can only be modified through the deposit and withdraw methods, ensuring data integrity. |