



VaultofCodes

JAVA INTERNSHIP

- OOPs in Java
- File Handling in Java

By

-Sarath Chandra Kagitha



OOPS IN JAVA

- ✓ Object-oriented programming System(OOPs) is a fundamental programming paradigm based on the concept of “objects”.
- ✓ These objects can contain data in the form of fields (often known as attributes or properties) and code in the form of procedures (often known as methods).

OOPs Concepts :

1. Class
2. Object
3. Method and Method Passing
4. Pillars of OOPs



CLASS

- ✓ A class is a user-defined blueprint or prototype from which objects are created.
 - ✓ Using classes, you can create multiple objects with the same behavior instead of writing their code multiple times.
-
1. **Modifiers:** A class can be public or have default access (Refer to this for details).
 2. **Class name:** The class name should begin with the initial letter capitalized by convention.
 3. **Superclass and Interfaces (if any):** The name of the class's parent (superclass), if any, preceded by the keyword `extends`. A class can only extend (subclass) one parent. A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword `implements`. A class can implement more than one interface.
 4. **Body:** The class body is surrounded by braces, { }.



OBJECT

- ✓ An object is a basic unit of Object-Oriented Programming that represents real-life entities.
- 1. **State:** It is represented by the attributes of an object. It also reflects the properties of an object.
- 2. **Behaviour:** It is represented by the methods of an object. It also reflects the response of an object to other objects.
- 3. **Identity:** It is a unique name given to an object that enables it to interact with other objects.



METHOD

- ✓ A method is a collection of statements that perform some specific task and return the result to the caller.
- ✓ A method is like a function i.e. used to expose the behavior of an object.
- ✓ It is a set of codes that perform a particular task.

Syntax :

```
<access_modifier> <return_type> <method_name>( list_of_parameters)
{
    //body
}
```



Advantages of Method :

- ✓ **Code Reusability:** Code reusability is a fundamental concept in software engineering that allows developers to reuse existing code in new applications
- ✓ **Code Optimization:** Code optimization is an essential aspect of software development that aims to improve the performance of a program.

Note :

- ✓ Methods are time savers and help us to reuse the code without retyping the code.



PILLARS OF OOPS

Pillar 1: Abstraction

- ✓ Data Abstraction is the property by virtue of which only the essential details are displayed to the user. The trivial or non-essential units are not displayed to the user.
- ✓ The trivial or non-essential units are not displayed to the user.

Pillar 2: Encapsulation

- ✓ It is defined as the wrapping up of data under a single unit. It is the mechanism that binds together the code and the data it manipulates.
- ✓ In encapsulation, the data in a class is hidden from other classes, which is similar to what data-hiding does.



Pillar 3: Inheritance

- ✓ It is the mechanism in Java by which one class is allowed to inherit the features (fields and methods) of another class. We are achieving inheritance by using extends keyword. Inheritance is also known as “is-a” relationship.

Pillar 4: Polymorphism

- ✓ It refers to the ability of object-oriented programming languages to differentiate between entities with the same name efficiently.
- ✓ As the names resembles *poly* refers to **many** and *morph* refers to **shape**, as a result *polymorphism* is a term which refers to **many shapes**.
- ✓ This is done by Java with the help of the signature and declaration of these entities. The ability to appear in many forms is called polymorphism.



FILE HANDLING IN JAVA

Java Files

- ✓ File handling is an important part of any application.
- ✓ Java has several methods for creating, reading, updating, and deleting files.

Java File Handling

- ✓ The File class from the java.io package, allows us to work with files.
- ✓ File Handling is an integral part of any programming language as file handling enables us to store the output of any particular program in a file and allows us to perform certain operations on it.
- ✓ In simple words, file handling means reading and writing data to a file.



STREAMS IN JAVA

- ✓ In Java, a sequence of data is known as a stream.
- ✓ This concept is used to perform I/O operations on a file.

Generating Streams

- ✓ `stream()` – Returns a sequential stream considering collection as its source.
- ✓ `parallelStream()` – Returns a parallel Stream considering collection as its source.



INPUT STREAM

Input Stream:

- ✓ The Java InputStream class is the superclass of all input streams. The input stream is used to read data from numerous input devices like the keyboard, network, etc. InputStream is an abstract class, and because of this, it is not useful by itself. However, its subclasses are used to read data.
- ✓ There are several subclasses of the InputStream class, which are as follows:
 1. ByteArrayInputStream
 2. AudioInputStream
 3. FileInputStream
 4. FilterInputStream
 5. StringBufferInputStream
 6. ObjectInputStream



OUTPUT STREAM

Output Stream:

- ✓ The output stream is used to write data to numerous output devices like the monitor, file, etc. OutputStream is an abstract superclass that represents an output stream. OutputStream is an abstract class and because of this, it is not useful by itself. However, its subclasses are used to write data.
- ✓ There are several subclasses of the OutputStream class which are as follows:
 1. ByteArrayOutputStream
 2. FileOutputStream
 3. StringBufferOutputStream
 4. ObjectOutputStream
 5. DataOutputStream
 6. PrintStream



FILE OPERATIONS IN JAVA

- ✓The following are the several operations that can be performed on a file in Java :

Create a File:

- ✓Create a File operation is performed to create a new file.
- ✓We use the `createNewFile()` method of file.
- ✓The `createNewFile()` method returns true when it successfully creates a new file and returns false when the file already exists.

Read from a File:

- ✓The next operation which we can perform on a file is "read from a file".
- ✓In order to write data into a file, we will use the Scanner class.
- ✓Here, we need to close the stream using the `close()` method.
- ✓We will create an instance of the Scanner class and use the `has NextLine()` method `nextLine()` method to get data from the file.



Write to a File():

- ✓ The next operation which we can perform on a file is "writing into a file".
- ✓ In order to write data into a file, we will use the `FileWriter` class and its `write()` method together.
- ✓ We need to close the stream using the `close()` method to retrieve the allocated resources.

Delete a File():

- ✓ The next operation which we can perform on a file is "deleting a file".
- ✓ In order to delete a file, we will use the `delete()` method of the file.
- ✓ We don't need to close the stream using the `close()` method because for deleting a file, we neither use the `FileWriter` class nor the `Scanner` class.



THANK YOU



Java

OOPs & File Handling



VaultofCodes