**Core Java - Assignment**

**Module - 1**

1. **Package and Access Modifiers**

**Theory : -**

1. Java Packages: Built-in and User-Defined Packages:

-> In Java, packages are used to group related classes and interfaces together. There are two types of packages in Java:

-> 1. Built-in Packages: These are provided by the Java API and include a wide range of classes and interfaces for various functionalities.

2. User-Defined Packages: These are created by developers to organize their own classes and interfaces.

1. Built-in Packages

-> Java provides a rich set of built-in packages that cover a wide range of functionalities. Some commonly used built-in packages include:

-> java.lang : Contains fundamental classes such as String, Math , Integer, System , and Thread .

-> java.util : Contains utility classes such as ArrayList, HashMap, Date , and Collections .

-> java.io : Contains classes for input and output operations such as File, InputStream , OutputStream, Reader , and Writer .

1. User-defined Package :

-> User-defined packages are created by developers to organize their own classes and interfaces.

For example. -> com.google.access;

2. Access Modifiers: Private, Default, Protected, Public :

-> Access modifiers are keywords that set the accessibility of classes, methods, and other members. There are four main access modifiers:

-> 1. Private: The member is accessible only within the same class.

-> 2. Default (no modifier): The member is accessible only within the same package.

-> 3. Protected: The member is accessible within the same package and by subclasses.

-> 4. Public: The member is accessible from any other class.

3. Importing Packages and Classpath.

-> Importing package :

-> To use classes from other packages, you need to import them using the import statement. You can import a single class or all classes from a package.

-> Setting the Classpath :

-> The classpath is a parameter that tells the Java Virtual Machine (JVM) and Java compiler where to look for user-defined classes and packages.

->You can set the classpath using the -classpath or -cp option.

project/

├── src/

│ ├── com/

│ │ ├── example/

│ │ │ ├── vehicles/

│ │ │ │ └── Car.java

│ │ │ └── main/

│ │ │ └── Main.java

└── bin/