**Assignment: (Core Java)**

**6 : Object-Oriented Programming (OOPs) Concepts**

**Que.1 Basics of OOP: Encapsulation, Inheritance, Polymorphism, Abstraction**

**Ans.1**  **Encapsulation:** Encapsulation is defined as the process of wrapping data and the methods into a single unit, typically a class. It is the mechanism that binds together the code and the data. It manipulates. Another way to think about encapsulation is that it is a protective shield that prevents the data from being accessed by the code outside this shield.

**Inheritance:** Inheritance is an important pillar of OOP. It is the mechanism in Java by which one class is allowed to inherit the features of another class. We are achieving inheritance by using extends keyword. Inheritance is also known as "is-a" relationship.

**Polymorphism:** The word polymorphism means having many forms, and it comes from the Greek words poly (many) and morph (forms), this means one entity can take many forms. In Java, polymorphism allows the same method or object to behave differently based on the context, specially on the project's actual runtime class.

**Abstraction:** Abstraction in Java is the process of hiding the implementation details and only showing the essential details or features to the user. It allows to focus on what an object does rather than how it does it. The unnecessary details are not displayed to the user.

**Que.2 Inheritance: Single, Multilevel, Hierarchical**

**Ans.2** Different types of inheritance which are supported by Java listed below

* Single Inheritance
* Multilevel Inheritance
* Hierarchical Inheritance
* Multiple Inheritance
* Hybrid Inheritance

**Single Inheritance:** In single inheritance, a sub-class is derived from only one super class. It inherits the properties and behavior of a single-parent class. Sometimes, it is also known as simple inheritance.

(Base class)

A

B

(Derived class)

**Multilevel:** In Multilevel Inheritance, a derived class will be inheriting a base class and as well as the derived class also acts as the base class for other classes.

A

(Base class 1)

(Base class 2)

B

(Derived class)

C

**Hierarchical:** In hierarchical inheritance, more than one subclass is inherited from a single base class. i.e. more than one derived class is created from a single base class.

(Base class)

Fruit

Banana

Apple

(Derived class)

**Que.3 Method Overriding and Dynamic Method Dispatch**

**Ans.3** Method overriding and dynamic method dispatch are core concepts in Java's object-oriented programming, enabling runtime polymorphism.

**Method Overriding:** Method overriding occurs when a subclass provides a specific implementation for a method that is already defined in its superclass. The overridden method in the subclass must have the same name, parameters, and return type as the method in the superclass. This allows a subclass to specialize or change the behaviour of an inherited method, providing a more specific implementation relevant to its own context. The Override annotation can be used to explicitly indicate that a method is intended to override a superclass method, which helps in compile-time error checking.

**Dynamic Method Dispatch (Runtime Polymorphism):** Dynamic method dispatch, also known as runtime polymorphism, is the mechanism by which Java determines which version of an overridden method to invoke at runtime. This happens when a superclass reference variable refers to an object of a subclass (known as upcasting). When the overridden method is called using this superclass reference, the Java Virtual Machine (JVM) dynamically determines which specific implementation of the method to execute based on the actual type of the object being referred to, not the type of the reference variable. This allows for flexible and extensible code, as the same method call can produce different behaviours depending on the actual object type at runtime.