

**Velammal College of Engineering and Technology, Madurai-625009**  
(Autonomous)  
**Department of Computer Science and Engineering**  
**Assignment I**

**Degree** : B.E/CSE **Year/Semester/section** : II/III/B  
**Course Code-Title** : 21CS203/Object Oriented Programming **Batch:** 2021-25  
**Name of the Instructor:** J.Shanthalakshmi Revathy  
**Announcement Date** : 18.09.22 **Submission Date** : 25.09.22  
**Total Marks** : 20 **Relevant COs** : CO1-K2

**Questions**

1. Relate OOPs concepts with real time Examples
2. What does the framework mean in Java? Explain in detail any one Java Framework.

**Instructions:**

- Submit softcopy with neat formatting -On time submission required
- Specify at least five references.
- Everyone should select individual examples and framework

**Evaluation Rubrics:**

Content	14
References	2
Presentation	2
Timely submission	2
<b>Total</b>	<b>20</b>

## 1. Relate OOPs concepts with real time Examples

- **Object:**

An entity that has state and behavior is known as an object e.g., chair, bike, marker, pen, table, car, etc. It can be physical or logical (tangible and intangible).

**An object has three characteristics:**

- **State:** represents the data (value) of an object.
- **Behavior:** represents the behavior (functionality) of an object such as deposit, withdraw, etc.
- **Identity:** An object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. However, it is used internally by the JVM to identify each object uniquely.

### **class**

A class is considered to be the template which is used to create a group of similar objects. In other words, a class defines what data the object will contain and how it will behave.

#### **syntax of class**

```
class <class_name>

{

    data members;    //hold the data or information

    member functions; //determine the behaviour

}
```

## **Real time examples for both class and object:**

**ClassName:** Cars

**DataMembers:** color,model,mileage

**Member Functions:** start, change gears, stop.

## **Inheritance :**

Inheritance in java is a core concept that requires the properties of one class to another class like a guardian. For example the relationship between father and son. Or also we can say that the properties derived from one class to another class are a term inheritance. The class inherits shared attributes and methodologies from another class. With the concepts of inheritance in java, the data in a program can be organized chronologically.

## **Types of inheritance :**

Single Inheritance

Multiple Inheritance

Multi-Level Inheritance

Hybrid Inheritance

Hierarchical Inheritance

### **Realtime example for inheritance:**

```
class A
{
int a, b;
void display()
{
System.out.println("Inside class A values =" +a+ " " +b);
}
}
class B extends A
{
int c;
void show()
{
System.out.println("Inside Class B values=" +a+ " "+b+ " "+c); }
}
class SingleInheritance
{
public static void main(String args[])
```

```
{  
B obj = new B(); //derived class object  
obj.a=10;  
obj.b=20;  
obj.c=30;  
obj.display();  
obj.show();  
}  
}
```

### **OUTPUT**

Inside class A values = 10 20

Inside class B values = 10 20 30

## **Polymorphism :**

The term “*polymorphism*” is an object-oriented programming term that means ability of a function, variable or object to assume different forms. The OOP languages that exhibit polymorphism allow functions to have same name but show different behaviours within same hierarchical tree structure (i.e. classes having same parent class.)

## **Realtime example :**

1. Another good real time example of polymorphism is water. Water is a liquid at normal temperature, but it can be changed to solid when it is frozen, or same water changes to a gas when it is heated at its boiling point. Thus, same water exhibiting different roles is polymorphism.
2. Even animals are great real life example of polymorphism, if we ask different animals to speak, they respond in their own way. Like if we ask Dog to speak it will bark, similarly, cow will moo, cat will meow. So the same action of speaking is performed in different ways by different animals exhibiting polymorphism.

## **Abstraction :**

Abstraction is one of the [key concepts](#) of object-oriented programming (OOP) languages. Its main goal is to handle complexity by hiding unnecessary details from the user. That enables the user to implement more complex logic on top of the provided abstraction without understanding or even thinking about all the hidden complexity.

## **Realtime example :**

You need to know how to use your coffee machine to make coffee. You need to provide water and coffee beans, switch it on and select the kind of coffee you want to get.

The thing you don't need to know is how the coffee machine is working internally to brew a fresh cup of delicious coffee. You don't need to know the ideal temperature of the water or the amount of ground coffee you need to use.

Someone else worried about that and created a coffee machine that now acts as an abstraction and hides all these details. You just interact with a simple interface that doesn't require any knowledge about the internal implementation.

You can use the same concept in object-oriented programming languages like Java.

## **Encapsulation :**

The process of binding data and corresponding **methods** (behavior) together into a single unit is called **encapsulation in Java**.

In other words, encapsulation is a programming technique that binds the class members (variables and methods) together and prevents them from being accessed by other classes.

## **Realtime example :**

1. When you log into your email accounts such as Gmail, Yahoo Mail, or Rediff mail, there is a lot of internal processes taking place in the backend and you have no control over it. When you enter the password for logging, they are retrieved in an encrypted form and verified, and then you are given access to your account. You do not have control over it that how the password has been verified. Thus, it keeps our account safe from being misused.

2. Suppose you have an account in the bank. If your balance variable is declared as a public variable in the bank software, your account balance will be known as public, In this case, anyone can know your account balance. So, would you like it? Obviously No.

So, they declare balance variable as private for making your account safe, so that anyone cannot see your account balance.

The person who has to see his account balance, will have to access only private members through methods defined inside that class and this method will ask your account holder name or user Id, and password for authentication.

Thus, we can achieve security by utilizing the concept of data hiding. This is called Encapsulation in Java.

## 2) What does the framework mean in Java?

**Java Framework** is the body or platform of pre-written codes used by Java developers to develop Java applications or web applications. In other words, **Java Framework** is a collection of predefined classes and functions that is used to process input, manage hardware devices interacts with system software. It acts like a skeleton that helps the developer to develop an application by writing their own code.

Framework are the bodies that contains the pre-written codes (classes and functions) in which we can add our code to overcome the problem. We can also say that frameworks use programmer's code because the framework is in control of the programmer. We can use the framework by calling its methods, inheritance, and supplying "callbacks", listeners, or other implementations of the Observer pattern.

### Hibernate framework

Hibernate is an open source object relational mapping ([ORM](#)) tool that provides a framework to map object-oriented domain models to relational databases for web applications.

Object relational mapping is based on the containerization of objects and the abstraction that provides that capacity. Abstraction makes it possible to address, access and manipulate objects without having to consider how they are related to their data sources.

The Hibernate ORM framework guides [mapping Java classes](#) to database tables and Java data types to SQL data types and provides [querying and retrieval](#).

### **Advantages of Hibernate**

Any changes made are encapsulated in the data source itself, so that when those sources or their application programming interfaces ([APIs](#)) change, the applications that use ORM don't have to make changes or even be aware of that information. Similarly, programmers can have a consistent view of objects over time, although the sources that deliver them, the sinks that receive them and the applications that access them may change.

Hibernate is freely available to download and is licensed under the open source [GNU Lesser General Public License \(LGPL\)](#).

## How does Hibernate work?

Hibernate is an open source Object-Relational Persistence and Query service for any Java Application. Hibernate maps Java classes to database tables and from Java data types to SQL data types and relieves the developer from most common data persistence related programming tasks.

Hibernate sits between traditional Java objects and database server to handle all the works in persisting those objects based on the appropriate O/R mechanisms and patterns.

## Hibernate and JPA vs. JDBC

[Java Database Connectivity](#) (JDBC) is an API packaged with the Java SE edition that makes it possible to standardize and simplify the process of connecting Java applications to external, relational database management systems (RDBMS).

Fundamentally, applications written in Java perform logic. The Java language provides facilities for performing iterative logic with loops, conditional logic with *if* statements and object-oriented analysis through the use of classes and interfaces. But Java applications do not store data persistently.

Data persistence is typically delegated to NoSQL databases such as MongoDB and Cassandra, or to relational databases such as IBM's DB2 or Microsoft's SQL Server or the popular open source database MySQL.

To help address the object-relational impedance mismatch, a number of frameworks exist that simplify the task of moving data [between a relational database](#) and a Java program. Popular object-relational mapping (ORM) frameworks include Hibernate, TopLink and DataNucleus. While each framework has its own set of unique capabilities, all of them comply with the Java Persistence API standard, which is now part of the [Java EE/Jakarta EE](#) specification.





