MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Final Year Internship Report

on

Software Development Engineer Intern

Submitted By:

Rishabh Jain

0901CS181082

Faculty Mentor:

Mr. Arun Kumar

Assistant Professor, MITS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Software Development Engineer Intern

A final year internship report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Rishabh Jain

0901CS181082

Internship Faculty Mentor:

Ms Jayati Munot

Learning And Development Persistent System

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

Persistent

PSL/HR/Cert-Add/2022
May 11, 2022

TO WHOMSOEVER IT MAY CONCERN

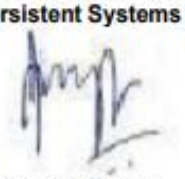
This is to certify that **Mr. Rishabh Jain** (Employee Code **47196**) is employed with us since **12 January 2022**. His designation is **Intern**.

According to the office records, his local and permanent addresses are :

Local Address	**Permanent Address**
5 Ex Anupam Nagar, Near Jiwaji University City Center Gwalior, Gwalior - 474011 Madhya Pradesh - India	5 Ex Anupam Nagar, Near Jiwaji University City Center Gwalior, Gwalior - 474011 Madhya Pradesh - India

This certificate is being issued on his request as a proof of employment and residence For verification purpose.

For Persistent Systems Ltd.



Manisha Tapaswi
Senior General Manager - Human Resources

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Rishabh Jain** (0901CS181082) has submitted the Internship report titled SDE Intern of the work he has done under the mentorship of **Mr Arun Kumar**, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Mr. Arun Kumar
Faculty Mentor
Assistant Professor
Computer Science and Engineering

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

Scanned with CamScanner

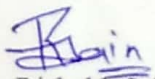
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this Internship report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in CSE at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Mr. Arun Kumar**, **Assistant Professor**, Department of CSE.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.



Rishabh Jain

0901CS181082

IV Year,

Computer Science and Engineering

Scanned with CamScanner

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

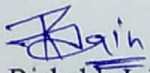
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester internship has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary internship as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this internship. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Mr. Arun Kumar**, Assistant Professor Department of Computer Science and Engineering, for his continued support and close mentoring throughout the internship. I am also very thankful to the faculty and staff of the department.



Rishabh Jain

0901CS181082

IV Year,

Computer Science and Engineering

Scanned with CamScanner

ABSTRACT

Persistent Systems is an Indian multinational technology services company which was incorporated on 16 May 1990 as Persistent Systems Private Limited. It work with many industry leading organizations across 18 countries - including 14 of the 30 most innovative US companies, 80% of the largest banks in the US and India, and numerous innovators across the healthcare ecosystem. In addition, Persistent Systems are recognized across the industry as the market leaders from delivery and customer excellence, as rated by ISG (2020 and 2021 ISG Star of Excellence Awards).Its head office is situated at Pune Maharashtra.

Persistent Systems provide a people centric work environment that enables our employees to:

- Accelerate growth, both professionally and personally.
- Impact the world in powerful, positive ways, using the latest technologies.
- Enjoy collaborative innovation, with diversity and work-life wellbeing at the core
- Unlock global opportunities to work and learn with the industry's best.

Persistent System Specialisation is Digital Transformation, Digital Business Strategy , CX Innovation & Optimization (SFDC), Digital Product Engineering (ISV), Data-Driven Business & Intelligence, Core IT Modernization , and Identity, Access and Privacy Management

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
GEMS	Graduate Engineers With Multidimensional Skills
VCS	Version Control System
GUI	Graphical User Interface
URL	Uniform Resource Locator
OOP	Object Oriented Programming
UML	Unified Modelling Language
DBMS	Database Management System
SQL	Structured Query Language
DML	Data Manipulation Language
DDL	Data Definition Language
DCL	Data Control Language
TCL	Transaction Control Language
JDK	Java Development Kit
JRE	Java Runtime Environment
JVM	Java Virtual Machine
POM	Project Object Model
JAR	Java Archive
STS	Spring Tool Suite
IOC	Inversion Of Control
MVC	Model View Controller
JDBC	Java Database Connectivity
AOP	Aspect Oriented Programming
ORM	Object Relation Mapping
API	Application Programming Interface

Chapter 1: INTRODUCTION

1.1 Internship Overview

Over the last 30 years, Persistent System have invested in building solutions, establishing partnerships, and getting the teams skilled to help their clients navigate these shifts and achieve business differentiation through elegant compositions of technology. The offerings and proven solutions create unique competitive advantage for the clients by giving them the power to see beyond and rise above.

Persistent System offered to me a six month internship program which is began from January 2022 to till July 2022. There is many module in internship cloud technologies like Salesforce , front end technologies like React Js , Angular , and some other technologies SDET,C++,Dot Net etc. These technologies are assigned to different students based on the business requirement.

So the module assigned to me is Core Java Track in which I have been trained about various skills like Git , Object Oriented System Design using UML and Core Java , Maven ,Spring and Spring boot. And some other non technical training like Power Skill and ELC(Experimental Learning Championship).

1.2 Objective and Scope

The objective of this Internship Program is to give exposure about various skill and technologies and cooperate culture and behaviour to the future Software Engineers which will be helping then to coordinate in the company and work with the assigned projects.

1.3 Internship Features

The Interns are trained on particular technology tracks assigned to them through the live lecture organised by the company through well renowned industry trainer. And after this training the performance of the candidate is examined by the assessments taken by the company which will be called as GEMS(Graduate Engineer with Multidimensional Skill) Assessments. Based on the performance and other factor like availability and business unit requirement the business unit and project is assigned to interns.

Chapter 2: Git

2.1 Git

In olden days developer has to face many problems while developing projects like if Some times the user performs some mistakes while working on git which results in the loss of information and he cannot revert back the changes and the developer face problem in managing the version of the projects and many more which can be solved by the Git which is a Version Control System (VCS).

Now we discuss that what is version control system.

The Version control system is a system that keeps record of specific files that you can get access to different versions of files easily with time . It helps the developer to work smarter and faster.

There are mainly 2 types of version control system:

- Centralized
- Distributed

Now we move on to the one of the most popular distributed version control system which is widely used by developer community in today's scenario.

Git is a most popular distributed version control system .It is a free and opensource tool. It has no central authority so changes can be shared without a server.

Git can be used in 2 ways :-

- Through command line
- Graphical user interface

Git is used in various workplaces some of them are :-

- marketing
- designers
- developers
- government

2.2 Git advantages

Following are the git advantages :-

- Git has good security and protection against the alteration of files.
- Git has good and faster network performance and superior disk utilization
- Git provide easy branching and merging
- Git three stage architecture
- Everything is done offline
- If no access to server or VPN, no need to wait till we get the access because everything is available locally if not take it from your friend (peer).

2.3 Git Architecture

Git mainly has 3 stages

- Working directory
- Staging area
- Git repository

2.3.1 Working directory:- The working directory is the folder in your local computer where the project files and folders are stored.

2.3.2 Staging Area:- The staging area has those files who are supposed to go to the next commit. Only those files which are needed to go to the next commit stay in the staging area.

2.3.3 Git Repository:- Git repo is a hidden file named .git. It stores all the commits and compresses them. So when you need a specific commit it can present that to you.

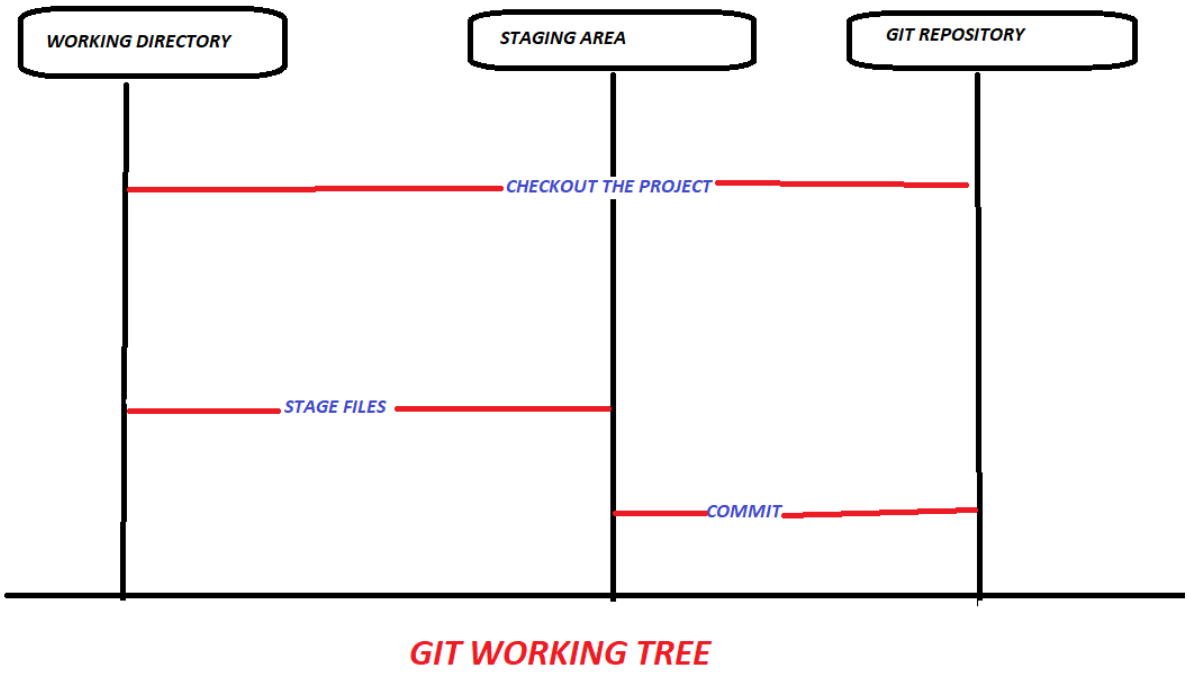


Fig 1

Now we move how to tracking our project

1. open the folder where your project or any file which you want to track is present
2. Right-Click anywhere while shifting and choose “Git Bash Here”
3. After Git bash is opened type “git status”.
4. It will give you a message that “fatal: not a git repository”.
5. It is a message that says these files are not considered as a git repository.
6. Now we will type “git init” to initialize this folder as a repository.
7. Now when we type the command “git status” then we will get the files which are present in the folder. Also, it says there that the files are not tracked.

8. Now to track all the files inside this got repository we will type “git add --a”, which will add all these files to the staging area.
9. Now we have to commit using this command “git commit -m “Initial Commit””.
10. Now if we do git status again it will say There’s nothing to commit, working tree clean. Which means we have successfully tracked our files.
11. Now to see the commits we have made, we will use “git log” command.
12. Now if we modify a single file and want to stage that only file we will use “git add filename.
13. Now we will commit with a message by typing “git commit -m “Your commit message””
14. Done Now You have successfully tracked your files.

2.4 Life Cycle of File in GIT

2.4.1 Untracked:- The files which are newly created in working directory and git does not aware of these files are said to be in untracked state.

2.4.2 Staged:- The files which are added to staging area are said to be in staged .These files are ready for commit.

2.2.3 In Repository/ Committed:- Any file which is committed is said to be In Repository/Committed State.

2.4.4 Modified:- Any file which is already tracked by git, but it is modified in working directory is said to be in Modified State.

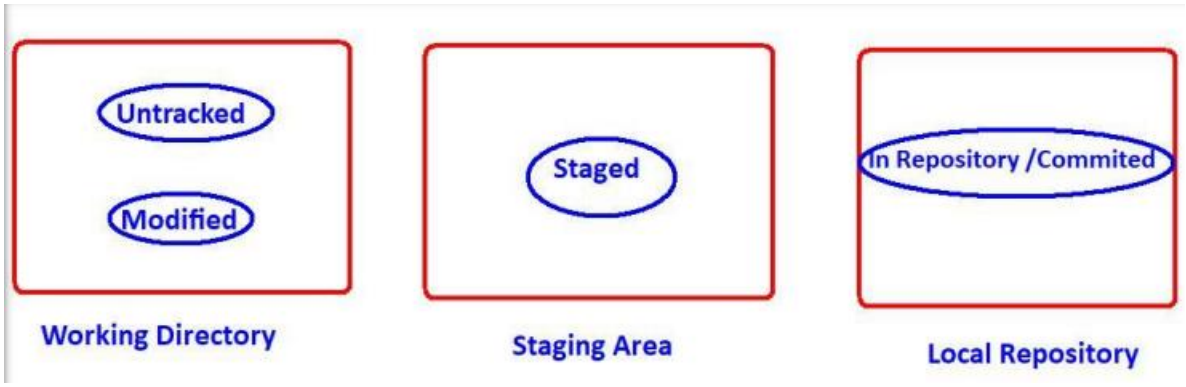


FIG 2 :- Git Lifecycle

2.5 Commands:-

These are common Git commands used in various situations:

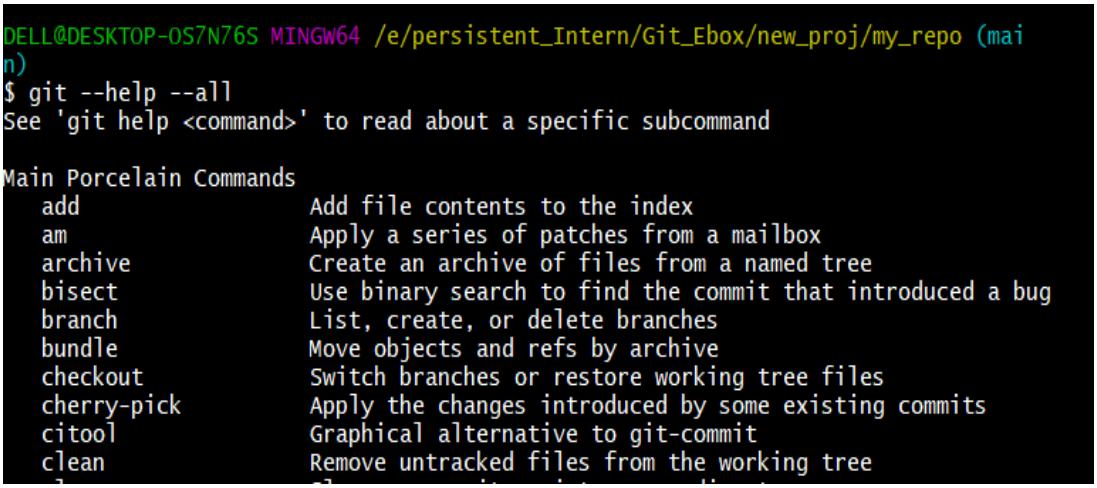
1. **git --version**

   It Prints the Git version that the git program came from.

```
DELL@DESKTOP-OS7N76S MINGW64 /e/persistent_Intern/Git_Ebox/new_proj/my_repo (mai
n)
$ git --version
git version 2.33.1.windows.1
```

2. **git –help:**

It prints the synopsis and a list of the most commonly used commands. If the option --all or -a is given then all available commands are printed.

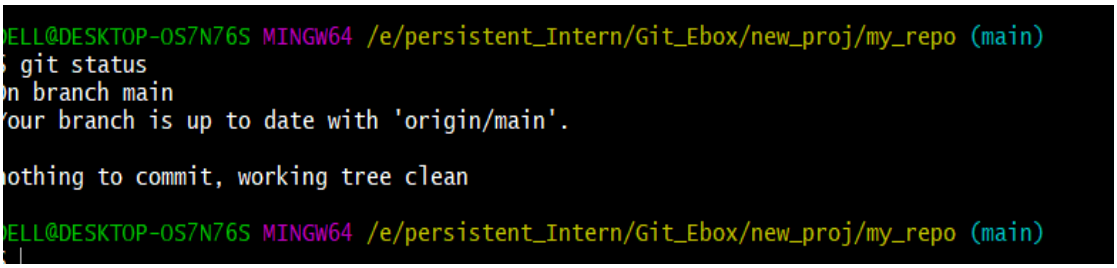


3. **git init:-**

Once we creates workspace, if we want version control, then we require a local repository. To create that local repository we have to use git init command

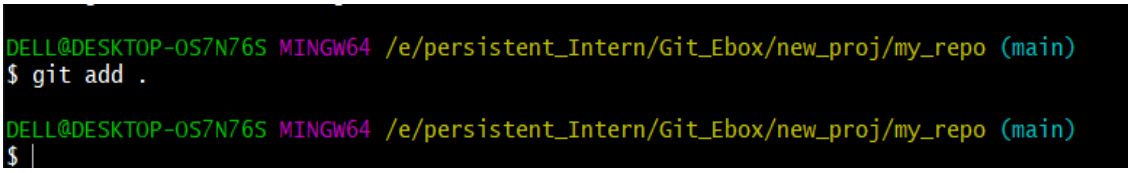
4. **git status:-**

It shows the current status of all files in each area, like which files are untracked, which are modified, which are staged etc



5. **git add:**

To add files from working directory to staging area for tracking/commiting purpose, we have to use git add command.



6. **git commit:**

If we want to commit staged changes, then we have to use git commit command. For every commit, a unique commit id will be generated. It is of 40-length hexadecimal string.

```
DELL@DESKTOP-OS7N76S MINGW64 /e/persistent_Intern/Git_Ebox/new_proj/my_repo (main)
$ git commit -m "commit"
[main 8e24f61] commit
 1 file changed, 1 insertion(+)

DELL@DESKTOP-OS7N76S MINGW64 /e/persistent_Intern/Git_Ebox/new_proj/my_repo (main)
$
```

7. **git log:**

It shows history of all commits. It provides commit id, author name,maild , timestamp and commit message.

• git log -n 3 → will display only 3 commits.

• git log –oneline → Condense each commit to a single line.

• git log --author= “ ” → Search for commits by a particular author. The argument can be a name of the author.

• git log --grep= “ ” → Search for commits with a commit message that matches , which can be a plain string or a regular expression.

```
DELL@DESKTOP-OS7N76S MINGW64 /e/persistent_Intern/Git_Ebox/new_proj/my_repo (main)
$ git log
commit 8e24f61e7cc0ddfefb1298df0b0f0833cba084be (HEAD -> main)
Author: Rishabh <jainrishabh2306@gmail.com>
Date:   Sat May 21 00:21:24 2022 +0530

    commit

commit adf96dfe289d439c6f59df80d7364cc1c6e774b8 (origin/main)
Author: Rishabh <jainrishabh2306@gmail.com>
Date:   Tue Feb 15 14:19:24 2022 +0530

    first commit

DELL@DESKTOP-OS7N76S MINGW64 /e/persistent_Intern/Git_Ebox/new_proj/my_repo (main)
$
```

8. **git config:**

This command is used to configure git like user name, mail id etc

git config --global user.email “rishabh_jain@persistent.com”

git config --global user.name "Rishabh Jain"

9. **git diff :**

git diff HEAD → This command shows difference in working directory and last commit.

git diff –cached → Show difference between staged changes and last commit

10. **git rm**

If we want to remove a file from working directory and from staging area then we should go for git rm command.

For e.g. git rm file1.txt

2.6 Working with Git Remote (Github):

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. GitHub is now the largest online storage space of collaborative works that exists in the world .It is the GUI to work with git.

Commands:-

- git remote add <name> <url>→Create a new connection to a remote repo. After adding a remote, you can use <name> as a shortcut for <url> in other commands.
- git fetch <remote> <branch> →Fetches a specific <branch>, from the repo. Leave off <branch> to fetch all remote refs.
- git pull <remote> →Fetch the specified remote's copy of current branch and immediately merge it into the local copy
- git push <remote> <branch> →Push the branch to <remote>, along with necessary commits and objects. Creates named branch in the remote repo if it doesn't exist.

CHAPTER 3- Object Oriented Programming

Object oriented Programming or oops which is a programming paradigm which is based on the concept of classes and objects, it will contain data in the form of fields and code in the form of procedure

3.1 Advantages/Disadvantages of oops:

Some of the advantages of oops are:-

- **Re-usability:** The one of the most important advantages of oops is reusability this feature help us to implement "write once and use anywhere" concept. We can achieve this using class.
- **Redundancy:** Redundancy is the problem where we have to write the same code again and again .Oops resolve this feature by using the concept of inheritance.

  For e.g. If two or more than two class have the same functionality so instead of writing the same functionality again and again in different class you can make the class which contain the common functionality and inherit that class to the other class so that the particular functionality come in all the class by writing only one time.
- **Easy Maintenance:** By writing the code with the oops paradigm it is easy to maintain the large programs.
- **Security:** Using data hiding and abstraction only necessary data will be provided thus maintaining the security of data

Some drawbacks of oops:

- **Size:** Object Oriented Programs are much larger than other programs.
- **Effort:** Object Oriented Programs require a lot of work to create.
- **Speed:** Object Oriented Programs are slower than other programs, because of their size.

3.2 Structure Or Building block in oops

The following are the building block of oops

- **Class:-** It is a user defined data type which act as a blue print for objects. Class are the logical abstraction. It help to encapsulate the data and procedure for form a single entity .It is declare with the help of class keyword.

  Syntax: class <class name>{ }
- **Objects:-** object is like an product of a class. Object can have state and behaviour .
- **Methods:-** Methods are the function that are defined inside the class that define the behaviour of the object
- **Attribute :-** It is also define inside the class template and represent the state of an object.

3.3 Pillers of oops:

Object oriented programming paradigm consist of 4 pillers :-

- Inheritance
- Abstraction
- Polymorphism
- Encapsulation

3.3.1 Inheritance:- The ability of creating a new class from an existing class. Inheritance allows a class (subclass) to acquire the properties and behaviour of another class (super-class). It helps to reuse, customize and enhance the existing code. So it helps to write a code accurately and reduce the development time.

For e.g. mobiles phone in today scenario is more advanced than the phone in olden days but todays phone have all the features which contain by the phone in olden days.

3.3.2 Abstraction :- Abstraction is the process of showing only essential/necessary features of an entity/object to the outside world and hide the other irrelevant information.

For e.g. to open your TV we only have a power button, It is not required to understand how infra-red waves are getting generated in TV remote control.

3.3.3 Polymorphism:- Polymorphism is combination of two word poly + morphs , the word poly means many and morphs means more than one so polymorphism is the ability to take more than one form. For e.g. a person is the employee at office customer at market parent at home etc.

3.3.4 Encapsulation:- Encapsulation means wrapping up data and member function (Method) together into a single unit i.e. class. Encapsulation automatically achieve the concept of data hiding providing security to data by making the variable as private

3.4 Class Diagram

Class diagrams are the important and the main building blocks of every object-oriented method. The class diagram can be used to show the classes, relationships, association, and the collaboration in the system. It describes the attributes and operations of a class and also the constraints imposed on the system.

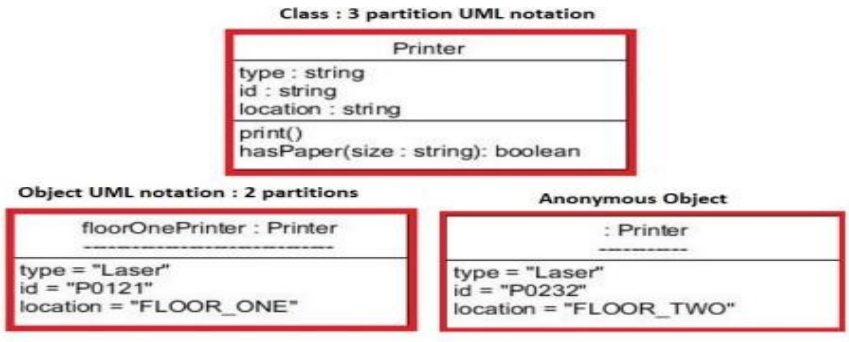


Fig 3 Class Diagram

3.5 UML

UML refer to the general purpose developmental modelling language that provide a standard way to visualize the design of a system. It help to model visualize construct and document software systems.

3.6 Relationship

One of the advantages of Object-Oriented programming language is code reuse. This reusability is possible due to the relationship b/w the classes.

- Association (Independent existence)
- Aggregation (Independent existence + container-contained)
- Composition (container-contained [does not exist without container])
- Inheritance ("Is A" relationship)

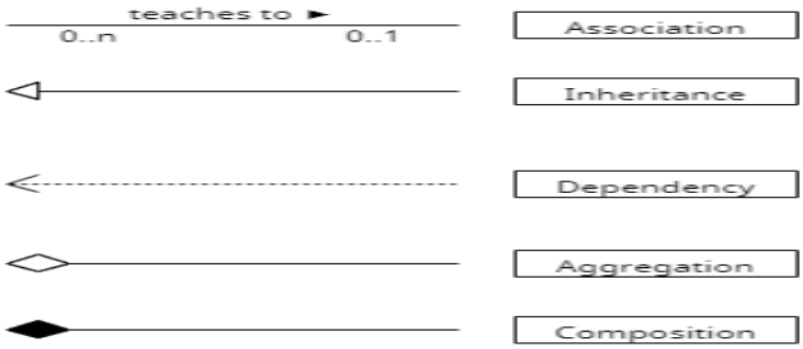


FIG 4- Relationships

3.6.1 Association

It is a link between objects of two or more class. An association exist between object of different class whether there is a logical linking or dependency between objects. Lifetime of a object is independent of each other.

For e.g. Company and employee has a relationship but both exist independently with each other.

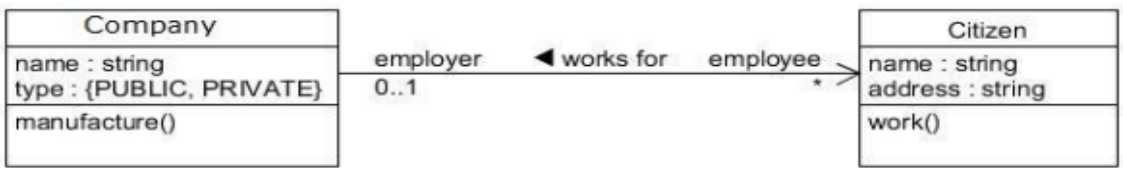


FIG 5- Association

3.6.2 Aggregation

It is whole/part relationship object which are composed of other object are known as aggregation .They may involve containment the contained object are component of larger object. Here container object can exist without contained object.

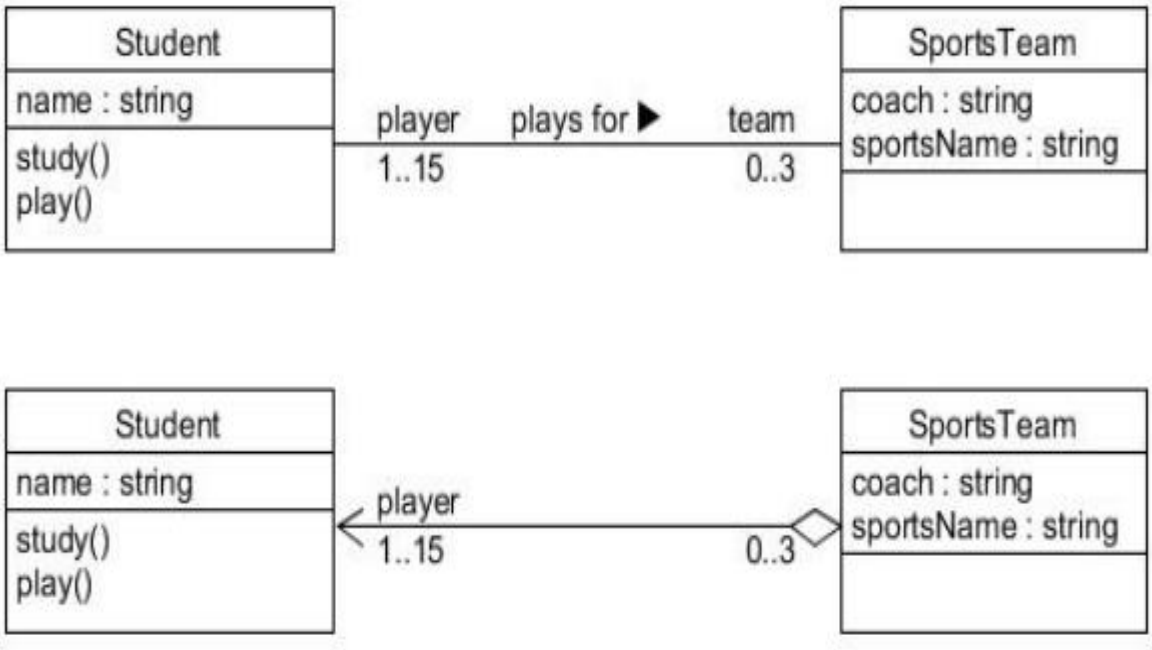


FIG 6- Aggregation

Aggregation Validation

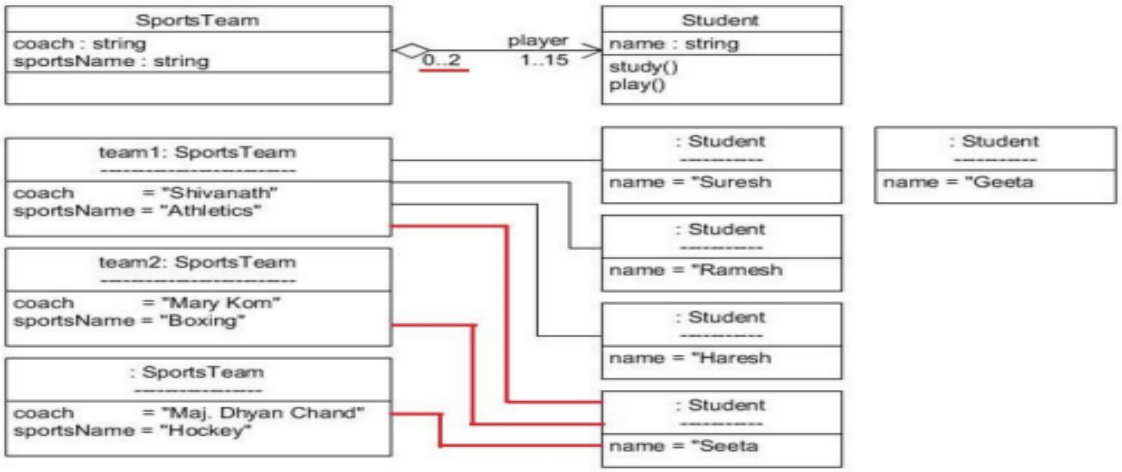


FIG 7 :- Aggregation Validation

3.6.3 Composition

It is a strong form of aggregation .It is of type owner/part .Here lifetime of a contained object is dependent of lifetime of container object ,i.e contained object cannot exist independently.

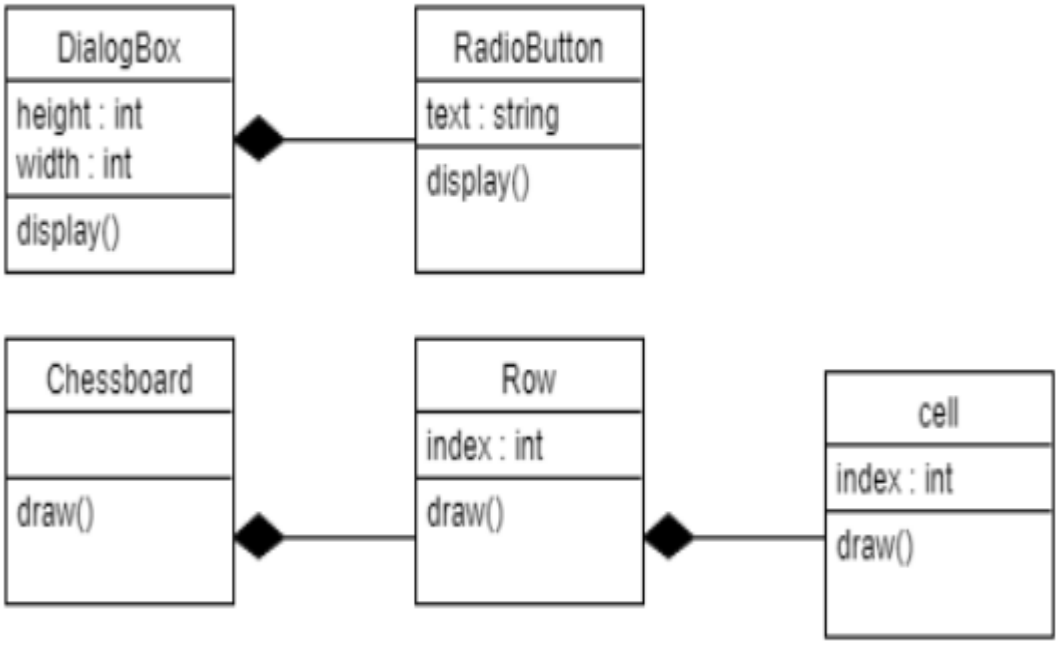


FIG 8- Composition

3.7 Feature scope/ visibility [UML]

In the UML class diagram we can also represent the visibility of the attribute like whether the attribute is private ,public ,protected or default.

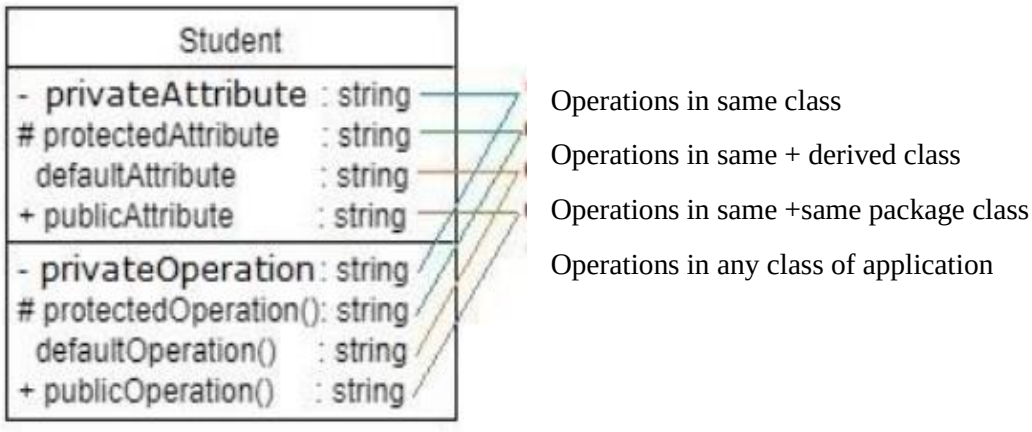


FIG 9-Scope Visibility

Q Draw the class diagram for the system consist of desktop and laptop?

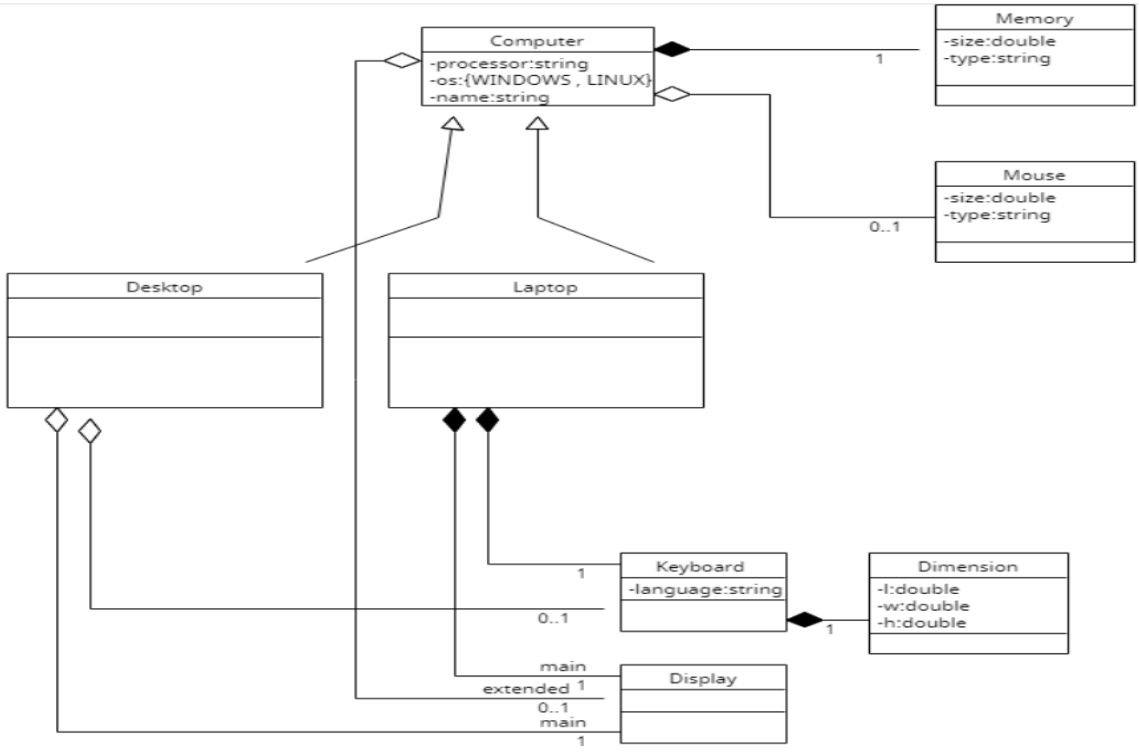


FIG 10-UML Diagram

Chapter 4: SQL

4.1 What is Database?

A database is simply an organized collection of related data, typically stored on disk, and accessible by many concurrent users. Databases are managed by a DBMS.A Database Management System (DBMS) is a set of programs that manages any number of databases.

In olden days for managing the data developer use file system but there are many problems while managing the data with file system. So the filesystem is replaced by the database management system.

Some of the advantages of DBMS are:-

- Reduced data redundancy
- Reduced updating errors and increased consistency
- Improve data integrity and application independence
- Improve data access for users by using host and query languages
- Improved data security
- Reduced data entry, storage, and retrieval costs
- Facilitated development of new applications program

As with every good some disadvantages are there. so some of the disadvantages of DBMS are:-

- Database systems are complex, complex and time consuming to develop
- Significant startup costs for hardware and software
- Database corruption affects almost all applications.
- Significant conversion costs when migrating from file systems to database systems
- Initial needs Training for all programmers and users

4.2 Entity Relationship Model:

In present scenario the most common database management system is Relational Database Management System refers as RDMS

- Entity:- An entity is a thing or an object in the real world that is distinguishable from other object based on the value of the attribute.
- Attribute:- It is the unit which describe the characterstics of entities. Some of the attribute in ER diagram are-
    - Simple Attributes.
    - Composite Attribute
    - Single Valued Attributes

- Multi-Valued Attributes
- Derived Attributes
- Stored Attributes

- Relationship:- It is an association between two or more entity of different entity set. It is mainly of 3 types;
    - One-to-One Relationship :- Each row in a table is related to a single row in another table. For e.g. One student can have one address. One address belongs to one student.
    - One-to-Many Relationship:- Each row in a table is related to multiple rows in another table. For e.g. One team contains multiple players. One player belongs to one team only.
    - Many-to-Many Relationship :- Each row in each table is related to multiple rows in another table. Each entity may be related to multiple occurrences of the other entity. For e.g. Project contains multiple employees and an employee can belong to one or more projects.

4.3 RDBMS:-

Data in an RDBMS is stored in database objects which are called as tables. This table is basically a collection of related data entries and it consists of numerous columns and rows.
Some Terminology.

- **Table:-** A Table is a basic storage structure of an RDBMS, It consisting of one or more columns and zero or more rows
- **Row:-** A Row is a combination of column values in a table. It is also referred to as RECORD or TUPLE.
- **Column:-** A Column represents one kind of data in a table. It is also referred to as ATTRIBUTE of an entity.
- **Field:-** At the intersection of Row and Column, we can find a Field. Field contains the actual data. If there is no data in the field, it is said to contain null value.
- **Primary Key:-** A Primary Key is a column or set of columns that it will uniquely identifies each row in a table. It must contain a value.
- **Foreign Key:-** A Foreign Key is a column or set of columns that refers to a primary key in the same or another table. A foreign key value must match the related existing primary key value or else be null.

4.4 SQL

SQL stands for Structured Query Language. It is a special purpose language designed for managing data in Relational database managements systems (RDBMS). It is a common language for variety of relational databases. It is basically used to communicate with a database.

So while managing data we can apply some constraint . Constraints are used to Enforce rules on the data in a table whenever a row is inserted, updated or deleted. The constraint must be satisfied for the operation to succeed.

The following are valid constraint types

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- DEFAULT

4.4.1 Types of SQL Statement:-

- **Data Definition Language (DDL):-** These statements are used to define the database structure or schema. They are used to manage different objects within the database such as Table, View, sequence, index, constraints etc for e.g. create ,alter etc
- **Data Manipulation Language(DML):-** These statements are required when user wants to add new rows to the table , Update existing rows or Remove rows from the table. That is when user want to manipulate the table.
- **Transaction Control Language(TCL):-** TCL commands are used to handle transactions in the database. These statements come in handy to manage and track the changes being made by the DML queries on our database. For e.g. commit , rollback
- **Data Control Language(DCL):-** It is mainly used for revoke and to grant the user the required access to a database . for e.g. grant , revoke.

4.4.2 Functions:

Functions are very powerful feature of SQL and can be used to perform operations on the data.

Some of the most widely used functions are:-

- Avg()
- Count()
- Max()
- Rank()
- Length()

4.4.3 Operator :-

In order to improve the functionality of condition there by to retrieve user required information we have to use the operator .There are various operator in Sql based of division like arithematic operator, wildcard operator etc some of them are:-

- Like operator:- It is used to determine whether a character column is satisfying specific style or not. The style can be verified by using some special symbols called "wild-card characters" . for e.g. %, _
- IN/NOT IN operator:- It is used to verify whether a column is equal or not equal to any one of the list of values specified or not. The values can be specified directly or by using a SELECT statement. For e.g. Select * from emp where sal IN(10000,20000,30000)
- UNION ALL: It is used to get combining data between two tables when the both tables are having same structure.
- INTERSECT: It is used to get the common data between queries.

4.4.4 Joins :-

A join clause is used to combine rows from two or more tables, based on a related column between them. In order to join the tables, they must have a common column. Common column means the names of the columns can differ, but they must have the same data type, size and data.

Types of Joins:

- Inner Join:- This join returns only the matching data between the tables. It is mainly used to retrieve the related data between the tables
- FULL Outer Join:- It retrieves all the rows; matched as well as unmatched from both the tables.
- Left Outer Join: It retrieves all the rows from the table which is at the Left hand Side(LHS) , even if there is no match in right table.
- RIGHT Outer Join:- It retrieves all the rows from the table which is at the Right hand Side (RHS), even if there is no match in left table.
- Self Join :- In this type of join we Join a Table to itself. Sometimes User needs to join table to itself. For example: To find employee's manager name, user need to join Employee table to itself. It is called as Self Join
- Cross Join:-: A join without common column criteria is called as a cross join. This join is used to find the different combinations of data between the tables. The output will be the product of number of records between the tables.

CHAPTER 5 -JAVA

5.1 JAVA

Java is one of the most popular and industry oriented high level programming language. Java was developed at Sun Microsystems by James Gosling Java is very much used by the industry due to its several good feature.

- **Platform Independent**

  It is one of the most popular feature of java .If we write Java program once, we can run it on any platform. So java follow WORA principle write once run anywhere.

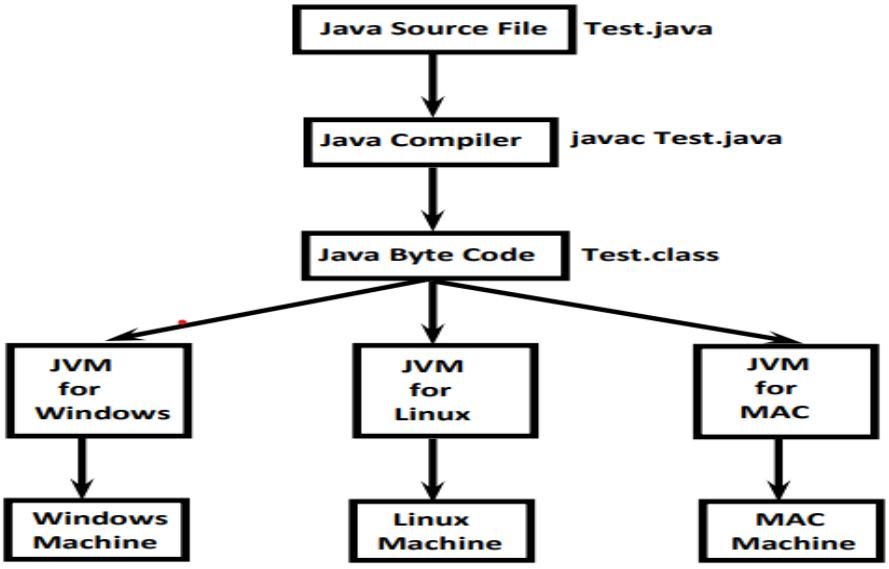


Fig 11 Platform Independent

- **Secure**

  The applications created by java is very much secure because the these programs never communicate directly with the machine .These programs is first converted into byte code and then jvm convert it to machine code.If the byte code contains any problem, then JVM did not permit that code to run and will raise VerifyError.

- **Object Oriented Programming Language**

  we all are familiar with this keyword so I did not go into the deep so Java is Object Oriented Programming Language like C++.Most of the times in java, we have to handle everything in terms of classes and objects

- **Multithreaded**

  It is Multithreaded i.e multiple threads it is the ability run same time and can perform specified tasks simultaneously, so that performance of the application will be improved.

- **Robust**

  Java is strongly typed language means Compiler will check each and every declaration and the assignments at compile time only for the type compatibility. If any problem w.r.t types, then at compile time only we can identify the problem Java provides Garbage Collector for automatic memory management. Hence there is no chance of memory related problems Java provides inbuilt Exception hanlding, which stop the abnormal termination of the program at runtime. Because of all these facilities, the chance of failing the program at runtime is very less and Hence Java is Robust

Some terminology

JDK (Java Development Kit) It provides environment to develop and run java applications.

JRE (Java Runtime Environment) It provides environment to run java applications

JVM (Java Virtual Machine) is an interpreter which is responsible to run java applications line by line.

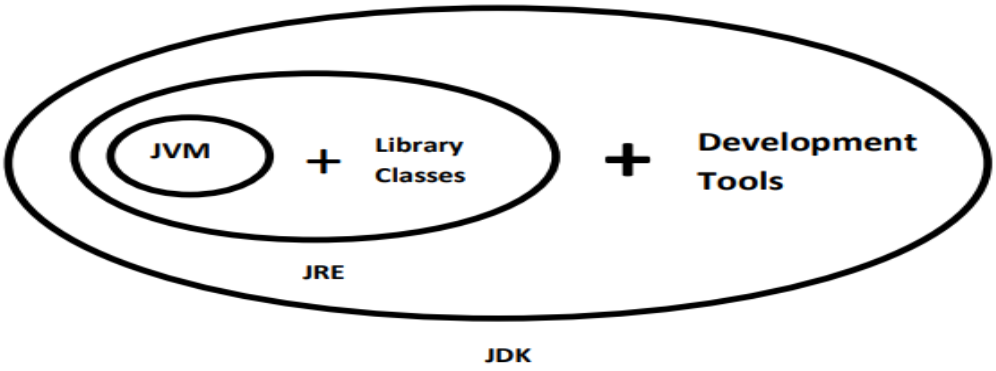


FIG 12:- JDK/JRE/JVM

5.2 Data Types

Every language has a data type it is used to store the data in the variable temporarily, for different different types of values there are different different types of datatypes like int float etc.Assignment should be checked by the compiler by the type compatibility hence java language is considered as strongly typed programming language..

Types of Variables:-

Based on the behaviour and position of declaration all variables are divided into the following 3 types

- Instance variables:-

  If the value of a variable is varied from object to object such type of variables are called instance variables i.e For every object a separate copy of instance variables will be created

- Static Variable:-

  If the value of a variable is not varied from object to object such type of variables is not recommended to declare as instance variables. We have to declare such type of variables at class level by using static modifier

- Local variables:-

  Some times to meet temporary requirements of the programmer we can declare variables inside a method or block or constructors such type of variables are called local variables.

5.3 Operator and assignments

Flow control:- Flow Control describes the Order in which the Statements will be executed at Runtime

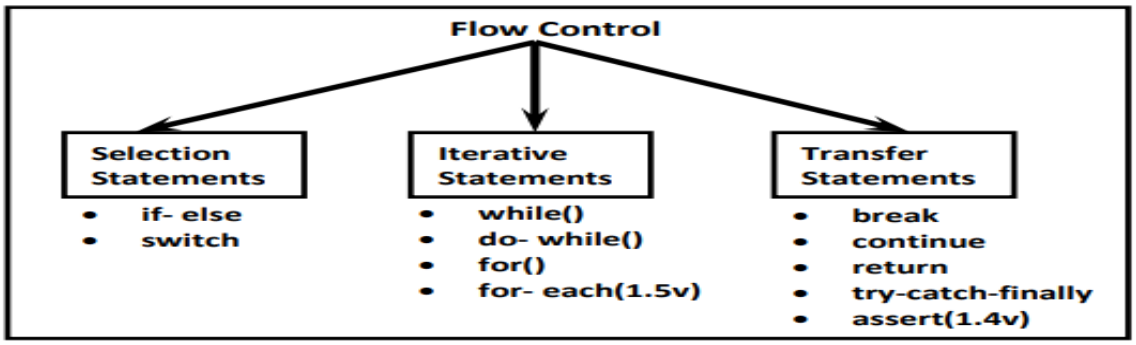


FIG 13:- FLOW CONTROL

5.4 Class Modifiers

Whenever we are writing our own classes compulsory we have to provide some information about our class to the JVM

Like

1) Whether this class can be accessible from anywhere or not.

2) Whether child class creation is possible or not.

3) Whether object creation is possible or not etc.

- **Public Modifier:-** If a class declared as public then we can access that class from anywhere. Within the package or outside the package

- **Default Modifier:-** If a class declared as the default then we can access that class only within the current package hence default access is also known as "package level access".

- **Final Modifier:-** Final is the modifier applicable for classes, methods and variables .Whatever the methods parent has by default it is available to the child .If the child is not allowed to override any method, that method we have to declare with final in parent class. That is final methods cannot overridden.

If a class declared as the final then we can't creates the child class that is inheritance concept is not applicable for final classes

for e.g.

```
final class Parent
{
}

class child extends Parent
{
}
```

OUTPUT:

Compile time error.

```
D:\Java>javac Parent.java
D:\Java>javac child.java
child.java:1: cannot inherit from final Parent
class child extends Parent
```

Every method present inside a final class is always final by default whether we are declaring or not. But every variable present inside a final class need not be final.

The main advantage of final keyword is we can achieve security.

The main disadvantage is we are missing the key benefits of oops

- **Abstract Modifier:-** Abstract is the modifier applicable only for methods and classes but not for variables. Even though we don't have implementation still we can declare a method with abstract modifier .That is abstract methods have only declaration but not implementation.

  Hence abstract method declaration should compulsory ends with semicolon

  Child classes are responsible to provide implementation for parent class abstract methods.

Member modifiers:-

- **Public members:-** If a member declared as the public then we can access that member from anywhere "but the corresponding class must be visible" hence before checking member visibility we have to check class visibility.
- **Private members:-** If a member declared as the private then we can access that member only with in the current class.
- **Protected members:** If a member declared as the protected then we can access that member within the current package anywhere but outside package only in child classes.

5.5 Interfaces:-

All methods inside an interface, whether declared or not, are always public and abstract.

Interface variables:-

Interfaces can contain variables. The main purpose of interface variables is to define requirement level constants. All interface variables, whether declared or not, are always public static and final.

For e.g.

```
interface One {
  void methodOne();
  }
  class Two {
  public void methodTwo() {
  }
  }
 class Three extends Two implements One {
  public void methodOne() {
  }
```

Interface vs Abstract Class

Interface	Abstract Class
If you don't have the implementation information and just have the sepecification you can move to interface .	If you have a partial implementation i.e. some method implementation you can go with abstract class

In the interface all methods whether declared or not are public and abstract	Every method may not be public and abstract.
Every method inside the abstract class need not necessary be public and abstract.	There are no restrictions on abstract class method modifiers.
Every interface variable whether declare or not always public static final	Not all abstract class variable need to be public static final
For required interface variables, initialization must be performed at declaration time. Otherwise, you will get a compile-time error.	It is not require performing initialization for abstract class variables at the time of declaration

Table 1-:Interface vs Abstract Class

5.6 OOPs

- **Data Hiding: -**According to it Our Internal Data should Not go out Directly OR Outside Person can't Access Our Internal Data Directly. This is the Concept of Data Hiding.
  e.g. After Providing Proper User Name and Password Only we can able to Access Our Mail Information.
  By declaring Data Member as private we can achieve Data Hiding.The Main Advantage of Data Hiding is Security.
- **Abstraction:-** Hiding Internal Implementation and Highlight the Set of Services which are offering is the Concept of Abstraction.
- **Encapsulation:-**It is the Process of Binding Data and Corresponding Methods into a Single Unit is Called Encapsulation.
  For e.g.

```
class Student {
 rollno;
 marks;
 name;
 age;
 +
 updateMarks()
 getName()
}
```

  In any Component follows Data Hiding and Abstraction Such Type of Component is

Called Encapsulated Component.

Encapsulation = Data Hiding + Abstraction

- **IS-A Relationship (Inheritance):-** Inheritance is the concept in oops which help us to make the reusabilty of code .By using extends Key Word we can implement IS-A Relationship There are mainly five type of inheritance but A Java Class can't extend More than One Class at a Time. Hence Java won't Provide Support for Multiple Inheritance with Respect to Classes because There May be a Chance of Raising Ambiguity Problems.

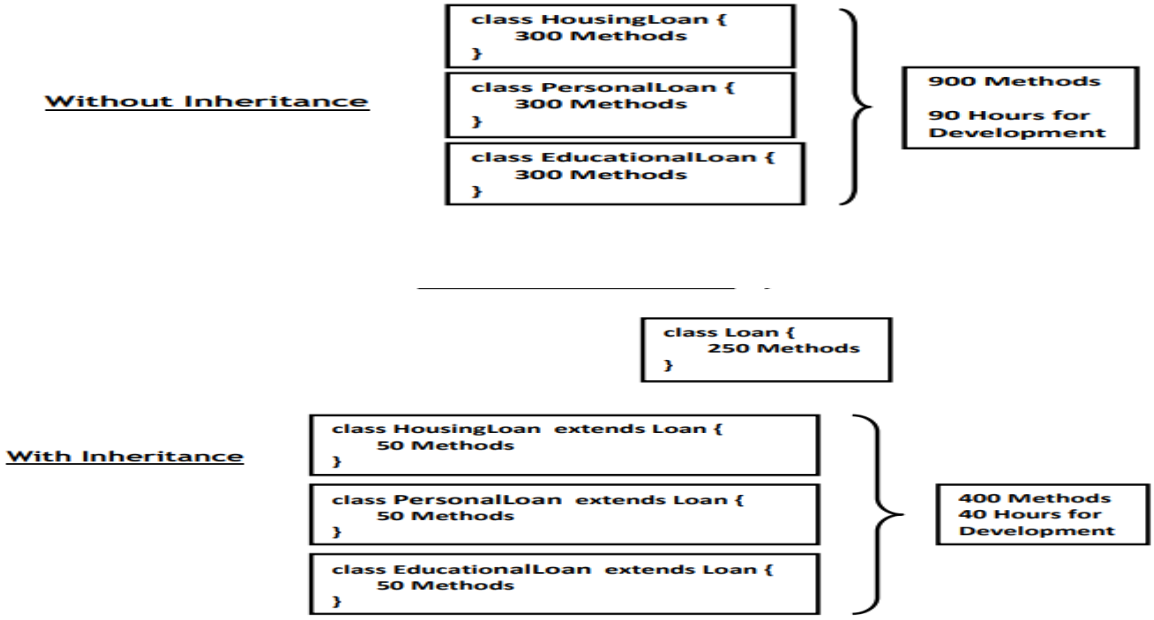


FIG 14-Inheritance

5.7 Exception Handling

It is the An Unwanted Unexpected Event that Disturbs Normal Flow of the Program is Called Exception. The Main Objective of Exception Handling is Graceful Termination of the Program. Exception Handling doesn't mean repairing an Exception. We have to define an Alternative Way to Continue Rest of the Program Normally .This Way of defining Alternative is nothing but Exception Handling.

For Example:- If Our Programming Requirement is to Read Data from the File locating at London. At Runtime if London File is Not Available then Our Program should Not be terminated Abnormally. We have to provide Some Local File to Continue Rest of the Program Normally. This Way of defining Alternative is nothing but Exception Handling.

In Our Program if all Methods Terminated Normally, then Only the Program will be Terminated Normally .In Our Program if at least One Method terminates Abnormally then the Program Termination is Abnormal Termination.

The exception are mainly divided into 2 types

- **Checked Exceptions:** The Exceptions which are Checked by the Compiler for Smooth Execution of the Program at Runtime are Called Checked Exceptions .Compiler Checks whether we are handling Checked Exceptions OR Not. If we are Not handling then we will get Compile Time Error
  For e.g. FileNotFoundException
- **Unchecked Exceptions:-** The Exceptions which are Not Checked by the Compiler are Called Unchecked **.** Exception .Compiler won't Check whether we are Handle OR Not
  For e.g Arithmetic Exception

In Java we can handle exception using the try and catch keyword

- **Try:-** In this block we put the risky code i.e the code in whice there is more chances of raising exception
- **catch:-** In this block we put the code where we put the code which is alternative code for the exception raised in try block.

It is Never Recommended to Define Clean-up Code Inside try Block. Because there is No Guaranty for the Execution of Every Statement Inside try Block .It is Never Recommended to Define Clean-up Code Inside catch Block. Because if there is No Exception then catch Block won't be executed.

Hence we required Some Place to Maintain Clean-up Code which should be executed Always irrespective of whether Exception raised OR Not raised and whether Handled OR Not Handled. Such Type of Best Place is Nothing but finally Block

Difference between final, finally and finalize

final: final is a Modifier is Applicable for Classes, Methods and Variables.If a Class declared as final then we can't Create Child Class. That is Inheritance is Not Possible for final Classes.If a Method declared as final then we can't Override that Method in Child Classes.If a Variable declared as final then we can't Perform Re Assignment for that Variable.

finally:- It is a Block Always associated with try-catch to Maintain Clean Up Code.The Specialty of finally Block is it will be executed Always Irrespective of whether Exception raised OR Not and whether Handled OR Not

finalize(): finalize() is a Method Always Called by the Garbage Collector Just before Destroying an Object to Perform Clean Up Activities .Once finalize() Completes Automatically Garbage Collector Destroys that Object.

Chapter 6 - Maven

Apache Maven makes life easier for programmers and developers. It is a software project management and understanding tool. We focused on the idea of the project object model (pom). Maven can handle project building, reporting, and documentation from a single trusted source. Help Java developers create and operate Java-based projects as well as day-to-day tasks.

The major purpose is to enable a developer to comprehend the entire state of a development endeavour in the lowest amount of time possible., Maven deals with several areas of concern:

- Making the build process easy
- Providing a uniform build system
- Providing quality project information
- Encouraging better development practices

The following are the key features of Maven is::

- It allows simple project setup which help to get a new project or module started in seconds
- Consistent usage across all projects - means no wastage of time for new developers coming onto a project
- Superior dependency management with including automatic updating, dependency closures.
- It help to easily manage with multiple projects at the same time.
- Dependency management: Maven supports managing dependencies using a central repository of JARs and other dependencies. Maven has a system that allows clients of a project to get the JAR files needed to build the project from a central JAR repository.

Terminologies In Maven

- **POM**: It stands for Project Object Model. An XML file that contains information about the project, dependencies that exist in the project, source file directories, plugin information, and many other files. This is the data Maven needs to fully build the project. Maven reads the pom file to get all this information.
- **GroupId**: Recognizes our project uniquely from all the projects. Pom file contains GroupId . It is often said as an identity for the group of projects.
- **ArtifactId**: A jar file that is deployed to the Maven repository. ArtifactId is a part of the pom file. It is often said as an identity and name of our project.
- **Version**: Specifies the version of the jar of the project. Version is also a part of the pom file.

6.1 Repositories

There are three types of repositories in Maven:-

6.1.1 Local Repository:

The local repository represents the developer's computer. All project materials are stored here. This repository contains all the dependency jars..

6.1.2. Remote Repository:

A remote repository refers to a repository on a web server used when Maven needs to download dependencies. This repository works in the same way as the central repository.Whenever something is needed from a remote repository, it is first loaded into the local repository and then used.

6.1.3. Central Repository:

Central repository refers to the Maven community that comes into action when there is a need for dependencies, and those dependencies cannot be found in the local repository. Maven downloads the dependencies from here whenever needed.

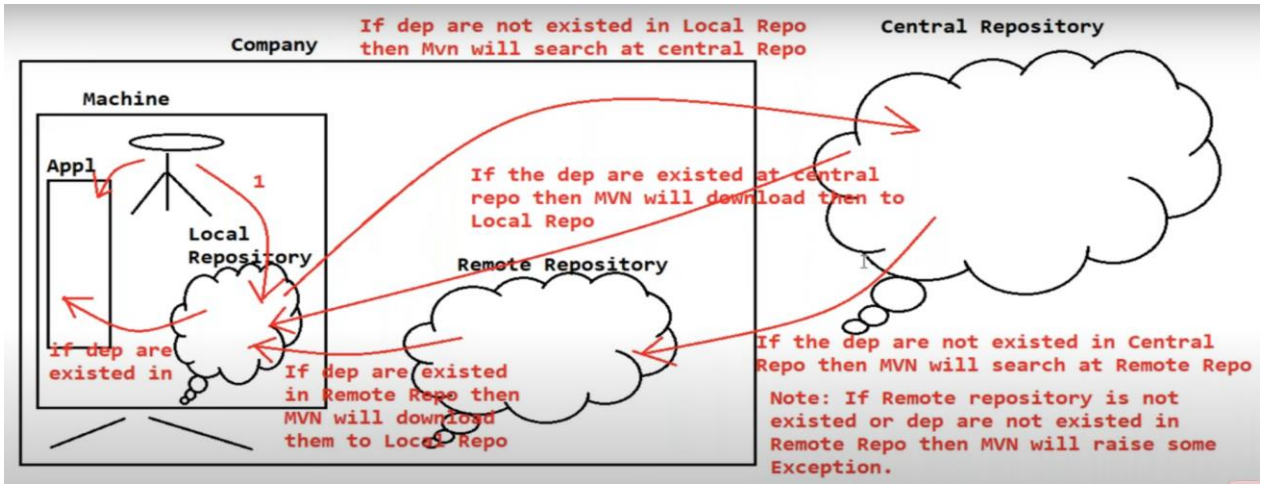


Fig 15 Repository

6.2 Build Life Cycles

Maven has 3 built-in build life cycles. These are:

- Default:- Takes care of the deployment of the project..
- Clean:- Removes any errors, thereby cleans the project, and removes the artifact produced from the previous process of the build.
- Site:- Takes care of the documentation of the project.

The default lifecycle has following some important phases:

- **Validate:** Verifies if all the prerequisites data for the build to complete are available.
- **Compile:** Source code is compiled.
- **Test- compile:** Test source code is compiled.
- **Test:** Unit test cases are executed.
- **Package:** Source code is compiled and packaged into JAR or ZIP files.
- **Integration- test:** Package is deployed and if there are any issues, integration test cases are executed.
- **Install-Package:** It is installed in the local repository.
- **Deploy:** A copy of the package is made available from the remote repository

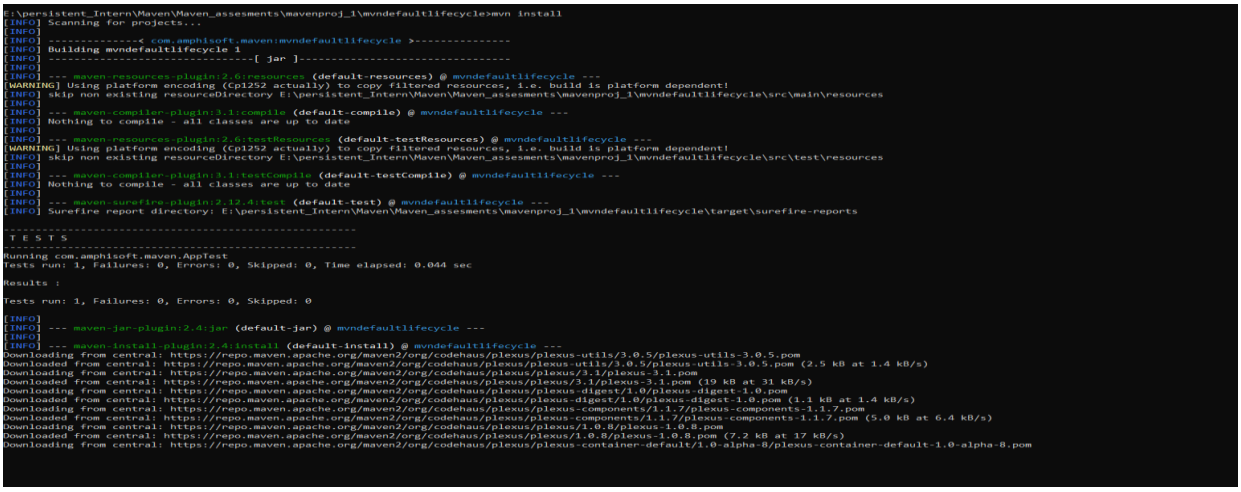
Some maven commands

- mvn validate:- This command validates the maven project that everything is correct and all the necessary information is available.

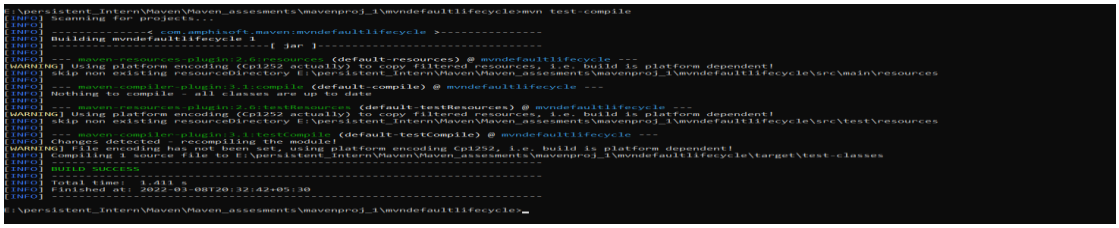
```
E:\persistent_Intern\Maven\Maven_assesments\mavenproj_1\mvndefaultlifecycle>mvn validate
[INFO] Scanning for projects...
[INFO]
[INFO] ---------------< com.amphisoft.maven:mvndefaultlifecycle >---------------
[INFO] Building mvndefaultlifecycle 1
[INFO] --------------------------------[ jar ]---------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.080 s
[INFO] Finished at: 2022-03-08T19:58:54+05:30
[INFO] ------------------------------------------------------------------------

E:\persistent_Intern\Maven\Maven_assesments\mavenproj_1\mvndefaultlifecycle>
```

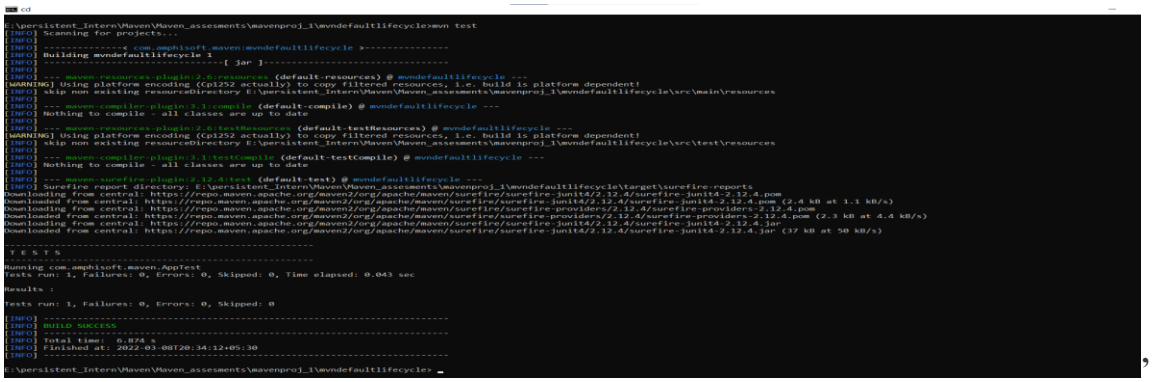
- mvn install:- This command builds the maven project and installs the project files (JAR, WAR, pom.xml, etc) to the local repository.



- mvn test-compile :- compile the test source code into the test destination directory.



- mvn test :- run tests using a suitable unit testing framework. These tests should not require the code be packaged or deployed.


,

- mvn clean: Cleans the project and removes all files generated by the previous build.
- mvn compile: Compiles source code of the project.
- mvn deploy: After compiling, running tests, and building the project, copy the packaged JAR/WAR files to the remote repository.

Chapter 7 -Spring

7.1 Spring

Spring is one of the most popular Java Enterprise Edition platforms. Spring is used by developers all over the world to build powerful, high-quality applications. It was designed by Rod Johnson in 2003. However, the main downside of the Spring project is that it is time consuming to set up and can be rather difficult for new developers

Enterprise is a business organization, it is a group of organizations running under single label.

An enterprise application is a software application that an enterprise prepares to simplify business .To prepare Enterprise Applications, we have to provide the following three layer

- User Interface Layer
- Business Processing Layer
- Data Storage And Access Layer

User Interface Layer:

This level is the highest level for enterprise applications. It provides users with a starting point for interacting with enterprise applications. This provides a very nice environment for receiving data from users and sending data to server-side applications..

Business Processing Layer:

It is the heart of enterprise applications and can be used to define and enforce all the business rules and regulations customers actually need. When developing an enterprise application to prepare the business processing layer, separate logic called "business logic" must be used, and to provide the business logic, technologies such as servlets, EJBs, and DAOs must be used.

Data Storage And Access Layer:

This tier is the lowest tier in an enterprise application and provides a very good environment for interacting with the database to perform storage operations. Preparing this layer in enterprise applications requires the use of separate logic called "store logic". To provide persistence logic, you need to use a technology set like JDBC, Hibernate, etc.

Features:-

- Loose coupling through dependency injection and interface orientation
- Aspects oriented programming
- Lightweight development with plain old Java objects (POJOs)
- Boilerplate reduction through aspects and templates

7.2 Core Module:

This is a core module of the Spring Framework. It provided the basis for all other spring-cage modules. This module can provide features such as IOC containers, beans, and dependency injection.

7.3 AOP Module[Aspect Oriented Programming]:

In general, if we prepare enterprise applications by using only Object Orientation then we have to provide both business logic and Services like Transactions, JMS, JAAS in combined manner, it will provide tightly coupled design, it will provide less sharability and less reusability.

7.4 JDBC/DAO Modules:

The main function of this module is to interact with database from Spring application in order to perform the database operations with JDBC .JDBC/DAO modules are able to abstract common JDBC implementation in order to simplify Database interaction from spring applications by providing template classes.

7.5 ORM Module[Object-Relational Mapping]:

ORM is the mapping between a table, properties or columns and primary key column with the respective bean component provided Bean class name, ID property, normal properties,ORM has define a set of rules and regulations to provide mapping between Object Oriented Data Model and Relational Data Model in order to achieve data persistency. E.g.: Hibernate, JPA,

WEB/WEB-MVC Modules:

WEB Module has provided very good environment to integrate other MVC based

framework applications like Struts, JSF. WEB-MVC is an MVC implementation provided by Spring framework directly in order to prepare web applications.

Steps to prepare First Spring Application

- Download Spring Framework from Internet.
- Provide Spring Setup in Eclipse/STS IDE
- Prepare Bean Class
- Prepare Bean Configuration File
- Prepare Test / Client Application.

Some Implementation:-

XML file for setting the xml configuration.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/X
        https://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/util
        https://www.springframework.org/schema/util/spring-util.xsd
        http://www.springframework.org/schema/context
        https://www.springframework.org/schema/context/spring-context.xsd">

    <!-- standalone list -->

    <context:component-scan base-package="com.springcore.annotations"></context:component-scan

    <bean class="org.springframework.jdbc.datasource.DriverManagerDataSource" name="ds" >
        <property name="driverClassName" value="com.mysql.cj.jdbc.Driver"></property>
        <property name="url" value="jdbc:mysql://localhost:3306/spring_yt"></property>
        <property name="username" value="root"></property>
        <property name="password" value="12345"></property>
    </bean>
    <bean name="jdbctemplate" class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="DataSource" ref="ds"></property>
    </bean>



    <!-- updated code -->
    <bean class="com.spring.jdbc.dao.Studentdaoimpl" name="studentdao">
     <property name="jdbctemplate" ref="jdbctemplate"></property>
    </bean>

</beans>
```

Design | Source

Problems | Javadoc | Declaration | Console | Coverage | Call Hierarchy | Servers

No consoles to display at this time.

Application file to run our program:-

```java
package com.spring.jdbc;

import org.springframework.context.ApplicationContext;

public class App {

    public static void main(String[] args) {
        System.out.println("hello world");
        System.out.println("my program started");
        ApplicationContext con=new ClassPathXmlApplicationContext("com/spring/jdbc/config.xml");
        //JdbcTemplate temp=(JdbcTemplate) con.getBean("jdbctemplate");
//      String q="insert into student values(?,?,?)";
        //int result=temp.update(q,456,"uttm_kumar","Kanpur");
        //System.out.println("no of record inserted"+result);


        //updated code
        Studentdao studentdao=(Studentdao) con.getBean("studentdao");

        Student student=new Student();
        student.setCity("lcknow");
        student.setId(100);
        student.setName("johan");
        int result=studentdao.insert(student);
        System.out.println("student addede "+result);



    }

}
```

Dao layer program:-

```java
package com.spring.jdbc.dao;

import org.springframework.jdbc.core.JdbcTemplate;

public class Studentdaoimpl implements Studentdao {

    private JdbcTemplate jdbctemplate;
    public int insert(Student student) {
        String q="insert into student values(?,?,?)";
        this.jdbctemplate.update(q,student.getId(),student.getName(),student.getCity());
        return 0;
    }
    public JdbcTemplate getJdbctemplate() {
        return jdbctemplate;
    }
    public void setJdbctemplate(JdbcTemplate jdbctemplate) {
        this.jdbctemplate = jdbctemplate;
    }


}
```

Our entity class:-

```java
package com.spring.jdbc.entities;

public class Student {

    private int id;
    private String name,city;


    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public Student(int id, String name, String city) {
        super();
        this.id = id;
        this.name = name;
        this.city = city;
    }
    public Student() {
        // TODO Auto-generated constructor stub
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getCity() {
        return city;
    }
```

Pom.xml file to manage dependency:-

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-ins
  <modelVersion>4.0.0</modelVersion>
  <groupId>Spring_Jdbc</groupId>
  <artifactId>Spring_Jdbc</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Spring_Jdbc</name>
  <description>Spring_Jdbc Application</description>

<dependencies>
    <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.3.14</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>5.3.14</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>5.3.14</version>
</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
```

7.6 IOC CONTAINERS

It is one of the most important principle in spring framework , it allow the classes in s spring to loosely coupled and make the programmer life easier to test and maintain them. It help to transfer the control of object and portion of the program to the framework.

There are mainly two type of IOC containers

- Bean Factory
- Application Context

Dependency Injection:- It is one of the main functionality provided by the spring framework .The spring -core module have the responsibility of injecting dependencies either by the constructor or getter setter method. Dependencies are passed in (injected) at service creation time. This is the situation when injecting elements of one class into another class. In practice, DI is implemented by passing parameters to constructors or using setters..

Goals of Dependency Injection

- Loose Coupling
- Coding to Interfaces
- Makes the code easy to understand, easy to test and maintain

Type of Dependency Injection:

- Setter Injection
- Constructor Injection
- Interface Injection

7.7 Spring Boot:

Spring Boot is a Spring framework module which provides RAD (Rapid Application Development) feature to the Spring framework. It is highly dependent on the starter templates feature which is very powerful and works flawlessly.

7.8 Spring Boot - Spring Boot starters

Spring Boot starters are templates that contain a collection of all the relevant transitive dependencies that are needed to start a particular functionality. For example, If you want to create a Spring WebMVC application then in a traditional setup, you would have included all required dependencies yourself. It leaves the chances of version conflict which ultimately result in more runtime exceptions.With Spring boot, to create MVC application all you need to import is spring-boot-starter-web dependency.

Spring Boot - Bootstrap the application

To run the application, we need to use @SpringBootApplication annotation. Behind the scenes, that's equivalent to @Configuration, @EnableAutoConfiguration, and @ComponentScan together .It enables the scanning of config classes, files and load them into spring context.

Advantages of Spring boot

- Spring boot helps in resolving dependency conflict. It identifies required dependencies and import them for you.
- It has information of compatible version for all dependencies. It minimizes the runtime classloader issues.
- It's "opinionated defaults configuration" approach helps you in configuring most important pieces behind the scene. Override them only when you need. Otherwise everything just works, perfectly.
- It helps in avoiding boilerplate code, annotations and XML configurations.
- It provides embedded HTTP server Tomcat so that you can develop and test quickly.

Chapter 8- Conclusion

The whole Idea of the Internship and training of selected candidates is to make them learn the cutting edge technologies and skills so that they can be assigned to projects and businesses in future as per the requirement. The candidates are supposed to learn and perform according to the standards set by the organisation to get the full time offer as well as for their future projects allocation. The candidates are trained for moulded to get together with the corporate environment and being useful for solving problems .During my Internship, I have completed all the mentioned modules in the above chapters. The assessments of the modules are still going on and I have cleared all of the assessments till now. There allocation of Projects will get initiated soon and candidates will be assigned to them according to the requirement of business units of the Organization.

REFERENCES

- **https://maven.apache.org/**
- **Persistent University**
- **https://spring.io**
- **https://docs.github.com/en**
- **https://www.baeldung.com/spring-boot**