MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

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

Final Year Internship Report

on

Java Full Stack

Submitted By:

Kanak Arya

0901CS181047

Faculty Mentor:

Dr. Rajni Ranjan Singh Makwana

Assistant Professor, CSE

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)

Java Full Stack

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

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Kanak Arya

0901CS181047

Internship Faculty Mentor:

Dr. Rajni Ranjan Singh Makwana

Assistant Professor, CSE

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

Internship Certificate :-

Persistent

PSL/HR/Cert-Add/2022
May 16, 2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. Kanak Arya** (Employee Code **47281**) 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**
201, Shri Apartment, Shanti Nagar Nai Sadak, Lashkar, Gwalior - 474001 Madhya Pradesh - India	201, Shri Apartment, Shanti Nagar Nai Sadak, Lashkar, Gwalior - 474001 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 **Kanak Arya**(0901CS181047) has submitted the Internship report titled **Software Development Engineering Intern** of the work he has done under the mentorship of **Dr. Rajni Ranjan Singh Makwana**, 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.

27/05/2022

Dr. Rajni Ranjan Singh Makwana

Assistant Professor
Computer Science and Engineering

27/05/22

Dr. Manish Dixit

Professor and Head,
Computer Science and Engineering

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

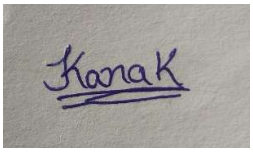
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(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 **Dr. Rajni Ranjan Singh Makwana**, Department of **Computer Science and Engineering**.

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



Kanak Arya
0901CS181047
IV Year,
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

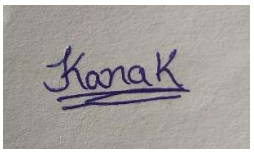
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

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, **Prof. Rajni Ranjan Singh Makwana**, 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.



Kanak Arya
0901CS181047
IV Year,
Computer Science and Engineering

ABSTRACT

Persistent Systems is an Indian multinational technology services company which was incorporated on 16 May 1990 as Persistent Systems Private Limited. Persistent Systems is listed on the Bombay Stock Exchange and the National Stock Exchange. 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 headoffice is situated at Pune Maharashtra .

Persistent Systems is a trusted digital engineering and enterprise modernization partner, combining deep technical expertise and industry experience to help our clients anticipate what's next and answer questions before they're asked. The offerings and proven solutions create unique competitive advantage for the clients by giving them the power to see beyond and rise above.

- We design and deliver new digital experiences, revenue streams and business models to meet rising customer expectations and accelerate their growth.
- We streamline your software engineering to drive greater efficiency and enable resiliency across your products and platforms.
- We Reinvent your applications, infrastructure and processes for greater agility by taking full advantage of automation, AI and cloud.
- We Transform and optimize the heartbeat of your operations, systems and product development to reduce costs and facilitate growth.

I am doing Internship in a Persistent System as a SDE Intern. My internship is duration is for 6 months and they assigned me a **Java track**. The internship is divided into 3 parts .

1. Month 1-4 : Study of some basic Learning modules and hands on .
2. Month 5 ; Advance Learning module based on BU.
3. Month 6 : project allocation.

In Month 1 they gives us 2 modules : Git , SQL.

Month 2 OOPS, Java Core

Month 3 Maven, Spring.

Month 4 Frontend Technology :- HTML,CSS ,JS.

And there is also some module vise test in every month and we do hands on these modules on E-Box company platfrom .

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: INTRODUCTION

1.1 About the Company

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 headoffice is situated at Pune Maharashtra.

Learning Objectives/Internship Objectives:

- Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training
- Internships in order to receive real world experience and develop their skills.
- An objective for this position should emphasize the skills you already possess in the area and your interest in learning more
- Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.
- Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
- Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position

1.2 My role in Internship

I am doing Internship in a Persistent System as a SDE Intern. My internship is duration is for 6 months and they assigned me a **Java track**. The internship is divided into 3 parts .

1. Month 1-4 : Study of some basic Learning modules and hands on .
2. Month 5 ; Advance Learning module based on BU.
3. Month 6 : project allocation.

In Month 1 they gives us 2 modules : Git , SQL.

Month 2 OOPS, Java Core

Month 3 Maven, Spring.

Month 4 Frontend Technology :- HTML,CSS ,JS.

Chapter 2: Git (Version Control System) :

2.1 Introduction to Git :

Git is a open source and distributed version control system which is designed to handle everything from small to very large projects with speed and efficiency.

Git has a tiny footprint with lightning fast performance and is easy to learn.

It outclasses SCM equipment(tools) like Subversion, CVS, Perforce, and Clear Case with functions like multiple workflows , convenient staging areas and cheap local branching.

Git comes with Integrated GUI tools (git-gui, gitk), however there are several third-party tools for users seeking for a platform--specific experience.

2.1.1 Benefits of Git :

•Git is distributed

• Most operations in Git only need local files and resources to operate, every operation in Git is local.

• Everything in Git is check-summed before it is stored i.e. It has integrity.

• The Three States – The Git working directory, The staging area and Repository.

• Everyone has the complete history.

• Everything is done offline.

• No central authority.

• Changes can be shared without a server.

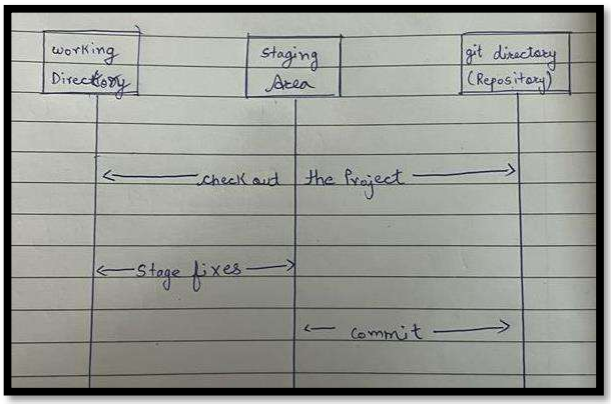


Fig-2.1.1.1 : Work flow of git

2.2 Version Control System(VCS) :

Version control system also called as source control, it is the practice of managing changes and tracking of the software code. Vcs are software tools manage changes to source code over time which help software teams. As development environments have accelerated, vcs help in work faster and smarter. They are especially **beneficial** for DevOps teams since they help them to increase successful deployments and reduce development time.

Vcs has a special kind of database in which they keeps track of every modification to the code .

2.3 Working with Git Local :

Following are the actions Git is able to Perform locally :-

- Creating local repository, adding files, and committing changes
- Viewing log and differences
- Staging changes as multiple changes
- Deleting and renaming files o Ignoring Files
- Undoing/redoing changes to the local copy and repository
- Cleaning the working copy

Git Commands:

git --version

Prints the Git suite model(version) that the git program got here from.

git –help

Prints the synopsis and a listing of the maximun used commands. If the option --all or **-**a is given then all to be had commands are printed.

2.4 Git Local Commands :

(i)Git init :

git init

this command is the first command that we use in Git. This command is used to creates a new blank repository and it is used to make existing project as a git project.

(ii)Git clone :

git clone <url>

this command can clone a repository from URL hosted location .

(iii)Git Config :

git config --global user.name “<firstname lastname>”

This command is used to set the user name of the user globally .

git config --global user.email “<valid-email>”

This command is used to set the email addfress of the user globally .

(iv)Git add :

git add <directory>

Stage all changes in <directory> for the next commit. Replace directory> with a <file> to change a specific file.

(v) Git Commit :

git commit -m *<message"

Commit the staged snapshot, but instead of launching a text editor, use <message> as the commit messag

(vi) Git Status :

git status

List which files are staged, unstaged, and untracked

(vii) Git log:

• git log -n 3 → this command display only 3 commits .

• git log –oneline → this command show each commit to a single line.

• git log --author=“” → Search for commits with a specific author. The argument may be a regular expression or normal string .

(vii)Git diff :

git diff HEAD → this command is used to Show difference between last commit and working directory.

git diff –cached → This command is used to Show difference between staged changes and last commit.

2.5 BRANCH & MERGE

git branch → This Command is used to get all the branch present in currently active bramch

git branch [branch-name] → This command will create the **branch [bramch-name]** locally in Git directory.

git checkout →this is used to switch to another branch in a repository.

git merge [branch] →this command is used to merge the current branch to specified branch’s in the repository.

2.6 Working with Git Remote (Github):

• GitHub is how developers(peoples) build software

• With a network(community) of million people, developers can use, discover, and contribute to projects using a effective collaborative development workflow.

• GitHub is used as code hosting platform for version-control system and collaboration.

• GitHub helps you and others developers to work together on Same projects from anywhere.

•Nowdays GitHub is the most important online storage space of collaborative works that withinside the world.

Commands :

git remote add <name> <url>→ this command is used to Create a new connection in a remote repository. After adding connection to a remote, you can make shortcut for <url> as <name>.

git fetch <remote> <branch> → this command is used to Fetches a specific <branch>, from the repository And leave off <branch> to fetch all the remote refernces.

git pull <remote> → this command is used to Fetch the Particular remote's copy of workin(current) branch and directly merge it into the local copy of repository.

git push <remote> <branch> → this command is used to Push the specific branch to <remote>, along with important objects and commits ,Also Create named branch in the remote repository if it does not exist.

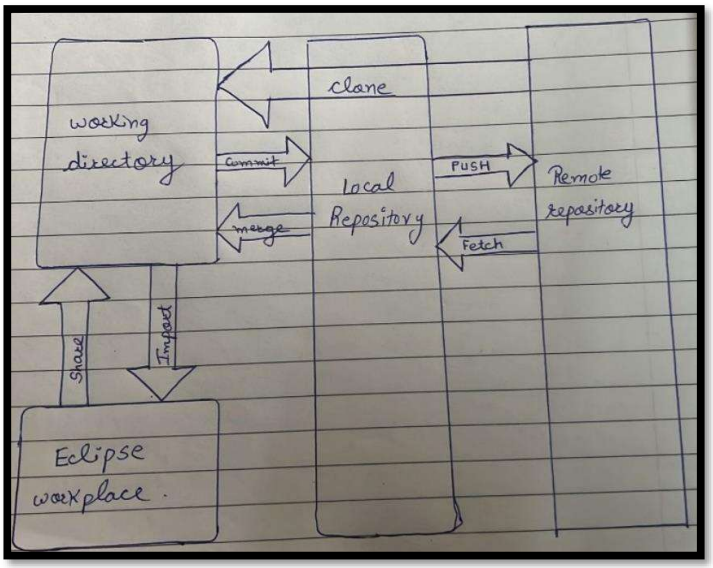


Fig 2.6.1: Working of local and remote Repository.

CHAPTER 3 : OOPS

3.1 UML Diagram

A UML diagram is a diagram primarly based totally at the UML (Unified Modeling Language) with the motive of visually representing a system,along with its main actors, actions, classes, artifacts or roles, in order to better understand, maintain, alter or document information about the system. Simplifies complicated(complex) software design, can also implement OOPs like a concept that is broadly used.

3.2 UML Class Diagram

The UML Class diagram is a graphical notation used to build(construct) and visualize object-orientated systems. A UML class diagram withinside the Unified Modeling Language (UML) is a type of static shape(structure) diagram that describes the structure of a system :

- classes,
- their attributes,
- operations (or methods),
- and the relationships among objects.

3.3 UML Class Notation

UML Class Notation is a class representation concept which encapsulated state (attributes) and behaviour (operations). Each operation(behaviour) has a signature. Each attribute(state) has a type. The class name is the only necessary(mandatory) information.

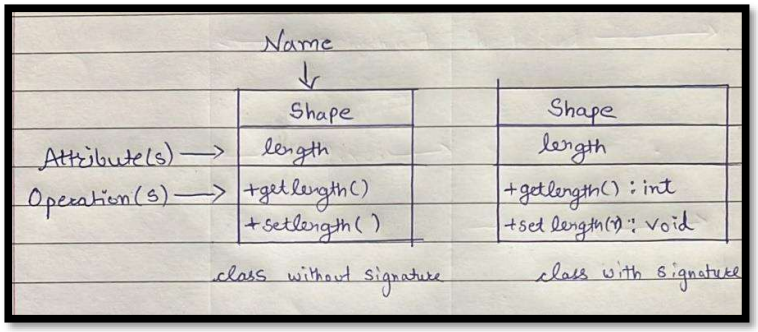


Fig 3.3 : Class Notation

3.3.1 Class Name:

- In UML Class Notation the first partition represent the class name.

3.3.2 Class Attributes:

- In the second partition of UML class notation Attributes are shown.
- The type of Attributes is shown after the colon symbol.
- Attributes map onto member variables (member function) in code.

3.3.3 Class Operations (Methods):

- In the third partition Operations are shown. They are services provides by the class.
- The return type of a method(operation) is shown after the colon symbol at the end of the method(operation) signature.
- The return type of method(operation) parameters is shown after the colon symbol following the parameter name. Operations map onto class methods(operation) in code

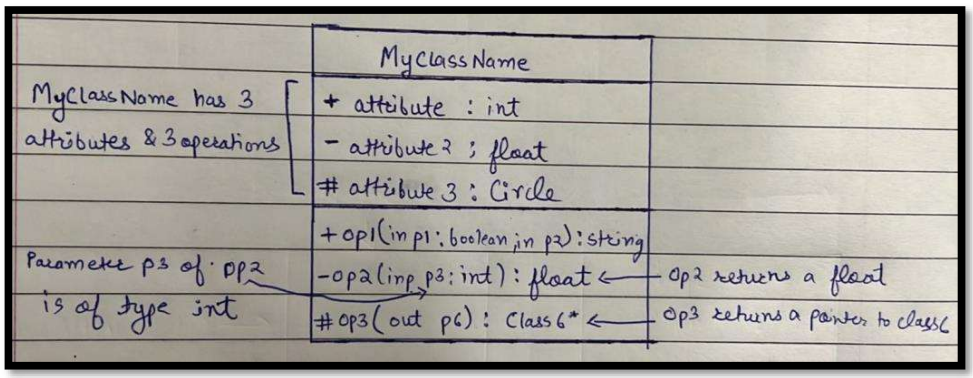


Fig 3.3.3 :- Class Operation

3.3.4 Class Visibility

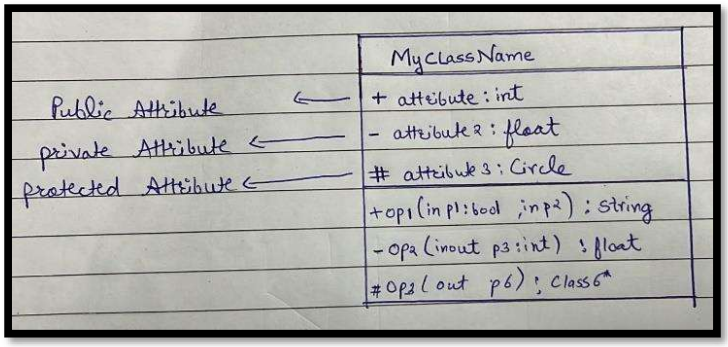


Fig 3.3.4 Class Visibility

- + denotes public attributes or operations

- - denotes private attributes or operations
- # denotes protected attributes or operations

3.4 Relationships between classes

UML isn't just about pretty pictures. If used efficiently, UML exactly(precisely) conveys how code should be implemented(applied) from diagrams. If precisely(exactly) interpreted, the carried out code will correctly reflect the intent of the designer. Can you describe what every of the relationships mean(imply) relative to your target(goal) programming language shown in the Figure below?

If you can not recognize them, no problem this section(segment) is supposed to help you to understand class relationships in UML. A class may be involved in single or multiple relationships with other classes.

the following type of relationships:

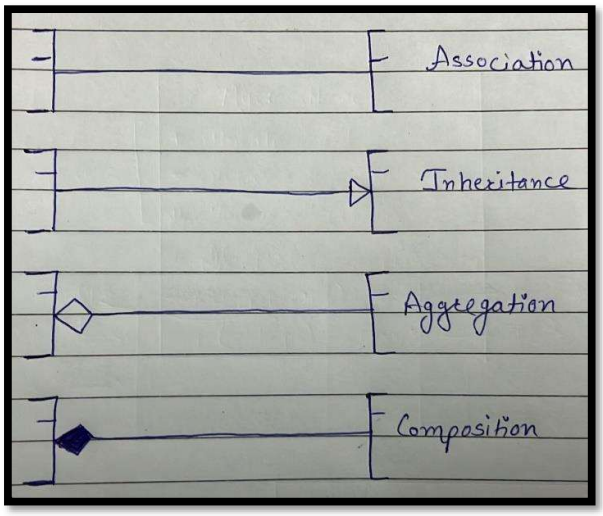


Fig 3.4 Types of relationship

3.4.1 Inheritance (or Generalization):

A generalization is a taxonomic relationship among a more general(wellknown) classifier and a more particular classifier. Each instance(object) of the specific classifier is also an indirect instance of the general classifier. Thus, the specific(particular) classifier inherits the features of the greater wellknown classifier.

- Represents an "is-a" relationship.
- An abstract class name is shown in italics.
- SubClass1 and SubClass2 are specializations of SuperClass.

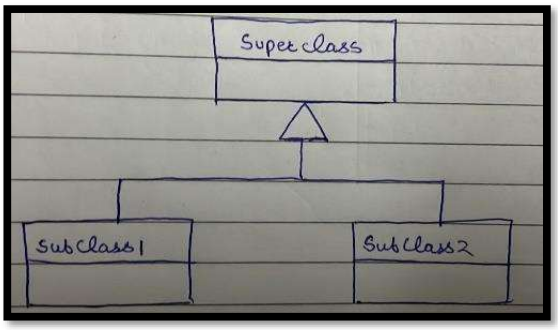


Fig 3.4.1 Inheritance

3.4.2 Association

In a UML Class Diagram Associations are relationships between classes.

Association are represented by a solid line between the classes.

Associations are generally named using a verb or verb phrase(word) which display(reflect) the real world problem domain.

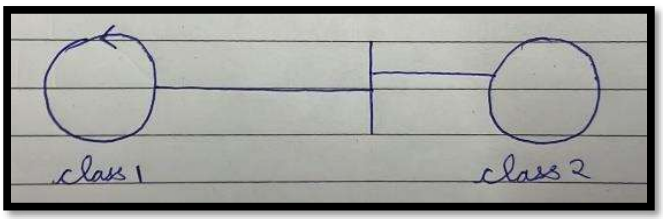


Fig 3.4.2 Association

Cardinality:

Cardinality is expressed in terms of:

- one to one
- one to many
- many to many

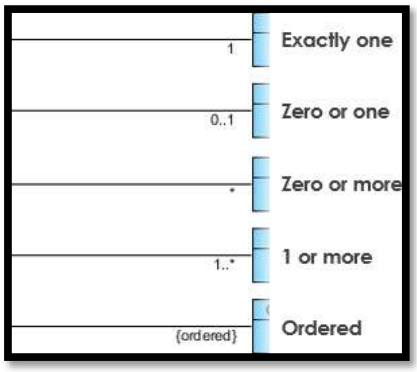


Fig 3.4.2.1 Cardinality

3.4.3 Aggregation

A aggregation is a special type of association.

- Aggregation represents a "part of" relationship.
- From below diagram we conclude that Class2 is part of Class1.
- Single or Many instances (denoted by the *) of Class2 can be associated with Class1.
- Both Class1 and Class2 Objects have separate lifetimes.

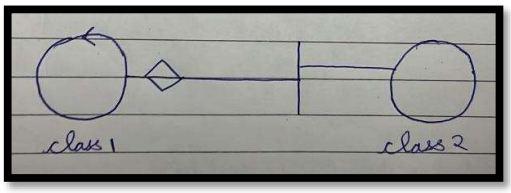


Fig 3.4.3 Aggregation

3.4.4 Composition

- Composition is a unique kind of aggregation in which components(parts) are destroyed while the Complete(whole) is destroyed.
- In Composition Objects of Class2 die and live(stay) with Class1.
- Class2 can't stand alone by itself.

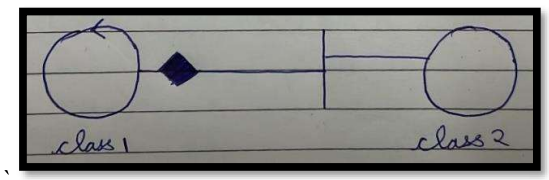


Fig 3.4.4 Composition

Chapter 4: SQL (STRUCTURED QUERY LANGUAGE)

4.1 what is 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.
- It is ANSI standard language; however there are variations that extend the ANSI subset. They are T-SQL, PLSQL, Sybase, Ingres, MySQL etc.

4.1.1 Advantages of SQL

• Efficient • Easy to learn and use • Scalable

• Functionality complete(With SQL, user can define, manipulate, retrieve data in the tables)

• Data integrity (SQL has many functionalities to maintain the integrity and consistency of data

4.2 Types of SQL Statements

1. DATA RETRIEVAL :- Retrieves data from the database

Statement :-

- SELECT.

2. DATA MANIPULATION LANGUAGE (DML) :- used to manage data within schema objects

Statement :-

- INSERT : insert data into a table
- UPDATE : updates existing data within a table
- DELETE : deletes records from a table, the space for the records remain.

3. DATA DEFINITION LANGUAGE (DDL) :- used to define the database structure or schema.

Statement :-

- CREATE :- to create objects in the database
- ALTER :- alters the structure of the database
- DROP :- delete objects from the database

4. TRANSACTION CONTROL (TCL) :- Used to manage the changes made by DML statements. Changes to the data can be grouped together into logical transaction.

Statement :-

- COMMIT : - To save work done
- ROLLBACK :- restore database to original since the last COMMIT
- SAVEPOINT :- identify a point in a transaction to which you can later roll back

5. DATA CNTROL LANGUAGE (DCL) :- Gives or removes access rights to both the oracle DB and structures within it.

Statement :-

- GRANT : - gives user's access privileges to database
- REVOKE :- withdraw access privileges given with the GRANT command

4.3 Joins in SQL

• When data from more than one table in database is required, a JOIN condition is used. Rows in one table can be joined to rows in another table according to common values existing in corresponding column, usually, primary and foreign key columns.

• Syntax:

SELECT table1.column, table2.column

From table1, table2

Where table1.column1=table2.column2; -- WHERE clause contains the condition that joins the tables together.

• Guidelines: - When writing the SELECT statement that joins the tables, precede the column name with table name for clarity and enhance database access. - If same column name appears in more than one table, column name must be prefixed with table name. - To join n tables, you need minimum of n-1 join conditions. This rule may not apply if your table has composite primary key.

4.3.1 Types of Join

There are 4 types of join:-

1. Inner join
2. Left outer join
3. Right outer join
4. Full outer join

These type of join are explain with the help of diagram given below:-

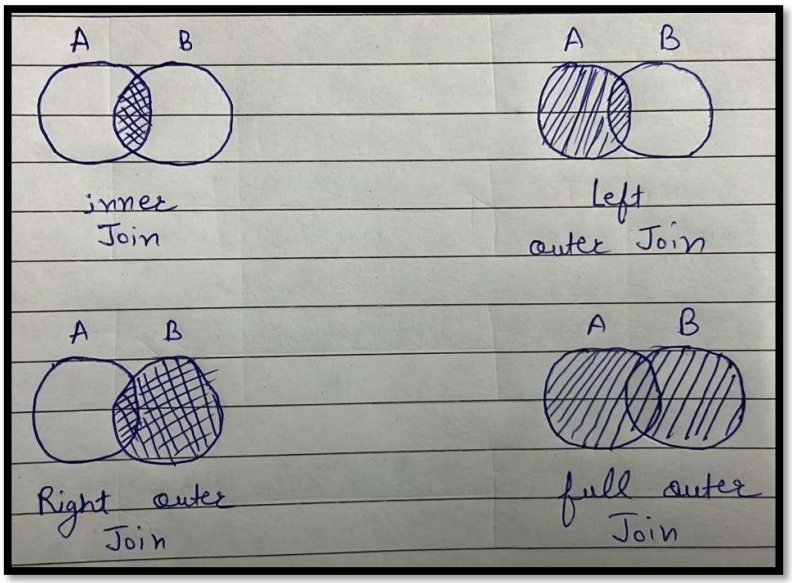


Fig 4.3.1 Types of joins

Chapter 5 : JAVA

5.1 What is Java?

Java is a platform in addition to a programming language. Java is a high-degree programming language this is additionally robust, object-oriented, and secure. Java changed into created in 1995 via way of means of Sun Microsystems (that's now a department of Oracle). James Gosling is famend as Java's father. It changed into referred to as Oak earlier than Java. Because Oak changed into already a recognized business, James Gosling and his group determined to adjust the call to Java. Platform: A platform is any hardware or software program surroundings wherein a programme runs. Java is known as a platform as it has a runtime surroundings (JRE) and API.

5.1.1 Java Features

- Easy to Learn
- Compiled and Interpreted
- Platform agnostic
- Portable
- Object-Oriented
- Secure
- Distributed
- Interactive and multi-threaded
- High Efficiency

5.2 Object oriented program (OOPS)

An Object-Oriented Program (OOPS) consists of a bunch of messages exchanging and co-operating between objects, to achieving a common objective.

Benefits of OOPs: -

- Real-global(world) programming
- Reusability of code
- Modularity of code
- Information hiding

5.2.1 Encapsulation

Encapsulation is the technique of hiding all the information of an item that don't make contributions to its crucial characteristics.

5.2.2 Abstraction

An Abstraction denotes the important traits of an item that distinguishes it from all different forms of gadgets and for that reason offers crisply described conceptual boundaries, relative to the attitude of the viewer.

5.2.3 Inheritance

• Inheritance is the functionality of a category to apply the residences and strategies of any other magnificence even as including its personal capability.

• Enables you to feature new capabilities and capability to an current magnificence with out editing the present magnificence.

• Superclass and Subclass

- A superclass or figure magnificence is the only from which any other magnificence inherits attributes and conduct.

–A subclass or infant magnificence is a category that inherits attributes and conduct from a superclass.

5.2.4 Polymorphism

• Derived from Latin words - Poly, this means that many, and morph, this means that forms. • It is the functionality of an movement or technique to do various things primarily based totally at the item that it's miles performing upon. • In item-orientated programming, polymorphism refers to a programming language's capacity to procedure items otherwise relying on their facts kind or class.

Two varieties of polymorphism are

• Compile time polymorphism • Runtime polymorphism

5.3 Class

• A class is a blueprint or prototype that defines the variables and the strategies not unusualplace to all gadgets of a sure kind.

• blueprint: A magnificence can not do something on its own.

• defines: A magnificence affords some thing that may be used later.

• objects: A magnificence(object) can best be used, if it had been "introduced to life" through instantiating it

5.3.1 Object

• An item is a software program assemble that encapsulates data, along side the capacity to apply or regulate that data, right into a software program entity.

• An item is a self-contained entity which has its very own personal series of properties (i.e. data) and methods (i.e. operations) that encapsulate capability right into a reusable and dynamically loaded structure.

5.3.2 Methods

• An operation upon an object, described as a part of the assertion of a class.

• The methods, described in a class, indicate, what the instantiated gadgets are capable of do.

5.3.3 Constructors

• A constructor is a unique(special) method that initializes the Instance variables. The approach

(method) implicitly when an instance(object) of class is created.

• A constructor approach has the identical call because the that of the magnificence.

• A constructor continually returns items of the magnificence kind as a result there's no go back kind laid out in its definition.

• A constructor is most usually described with the accessibility modifier "public" in order that each software can create an example however isn't always mandatory.

• A constructor is supplied to initialize the example variables withinside the magnificence whilst it's far referred to as.

5.3.4 this reference

• "this" represent the reference of the current object.

• This is used to - refer to the object who called it, when it must be surpassed as a parameter to a method.

5.4 Exception Handling

What is Exception Handling?

Exception Handling is a mechanism to deal with runtime mistakes including ClassNotFoundException, IOException, SQLException, RemoteException, etc.

5.4.1 Types of Java Exception

There are in particular kinds of exceptions: checked and unchecked. An blunders is taken into consideration because the unchecked exception. However, in step with Oracle, there are 3 kinds of exceptions namely:

1. Checked Exception
2. Unchecked Exception
3. Error

5.4.1.1 Checked Exception

The lessons that at once inherit the Throwable elegance besides RuntimeException and Error are referred to as checked exceptions. For example, IOException, SQLException, etc. Checked exceptions are checked at compile-time.

5.4.1.2 Unchecked Exception

The lessons(class) that inherit the RuntimeException are referred to as unchecked exceptions. For instance, ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException, etc. Unchecked exceptions aren't checked at compile-time, however they may be checked at runtime.4.6.1.3 Error .Error is irrecoverable. Some instance of mistakes are OutOfMemoryError, VirtualMachineError, AssertionError etc.

5.4.2 Java Exception Keywords

Java offers 5 key phrases which can be used to deal with the exception. The following table describes each.

Keyword	Description
try	try key-word is used to specify a block wherein we must location an exception code. It way we cannot use attempt block alone. The attempt block have to be observed through both seize or sooner or later.
catch	Catch is used to deal with the exception. It have to be preceded through attempt block this means that we cannot use seize block alone. It may be observed through sooner or later block later.
finally	finally is used to execute the important code of the program. It is performed whether or not an exception is treated or not.
throw	Throw key-word is used to throw an exception.
throws	Throws key-word is used to claim exceptions. It specifies that there can also additionally arise an exception withinside the technique. It does not throw an exception. It is usually used with technique signature.

Chapter 6: Maven

6.1 What is Maven

Apache Maven is a software program assignment control and comprehension tool. Based at the idea of a assignment item model (POM), Maven can manipulate a assignment's build, reporting and documentation from a primary piece of information.

6.1.1 Some Features:

• Simple assignment setup

• Consistent utilization throughout all tasks (project)

• Dependency control

• Able to without problems paintings with a couple of initiatives on the equal time

• Instant get admission to to new functions with very little more configuration

• Web web website online or PDF technology inclusive of any documentation

• Release control and distribution publication

6.1.2 Broader pictures

• Lifecycle - broadest unit

• Phase - A stage in the lifecycle.

• Goal - A plugin goal represents a specific task

6.1.3 Lifecycles in maven

• default - handles your project deployment

• clean - handles project cleaning

• site - handles the creation of your project's site documentation

6.1.4 default Lifecycle phases

• **validate -** validate the project is correct and all necessary information is available

• **compile -** compile the source code of the project

• **test -** test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed