MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

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

Final Year Internship Report

on

JAVA and Full stack Technology

Submitted By:

Akhil Umoriya

0901CS181004

Faculty Mentor:

Dr. Manish Dixit

Professor and Head,

Department of Computer Science and Engineering

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 and Full stack Technology

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:

Akhil Umoriya

0901CS181004

Internship Faculty Mentor:

Dr. Manish Dixit

Professor and Head,

Department of Computer Science and Engineering

Submitted to: Dr. Manish Dixit

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 17, 2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. Akhil Umoriya** (Employee Code **46829**) 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**
02-Goberdhan bihar, Char seher ka naka, M.P. B.Tech.-Computer Science and Engineering S/o, Gwalior - 474015 Madhya Pradesh - India	02-Goberdhan bihar, Char seher ka naka, M.P. B.Tech.-Computer Science and Engineering S/o, Gwalior - 474015 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 **Akhil Umoriya (0901CS181004)** has submitted the Internship report titled **Academic Engineer Intern** of the work he has done under the mentorship of **Jayati Munot**, 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.

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

22/05/20

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

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

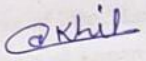
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(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. Manish Dixit**, **HOD**, 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.



Akhil Umoriya
0901CS181004
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 , **Dr. Manish Dixit**, **HOD**, 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.

Akhil Umoriya

0901CS181004

IV Year,

Computer Science and Engineering

ABSTRACT

In the present era technology has been the main concern because it is the one which connect one to other and has changed the lives of our and for this we need software and applications to run on platform. Here I have assigned as Software Engineer Intern. My Internship is basically divided into two phases phase 1 and phase 2 and right now my phase-1 training is going on in which I have learnt about different modules which form the key components to build any software and then in phase-2 training I will undergo the process to learn advance module and then the project will be allocated.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: INTRODUCTION

ABOUT THE COMPANY

Persistent Systems is an Indian multinational technology services company which was found in 1990 by Anand Deshpande. The current CEO of the company is Sandeep Kalra. Headquarters are located in Pune. It is providing it's services in Digital Strategy and Design, software product Engineering, Cloud & Infrastructure and many more to different domains life science, healthcare, financial , telecom & Media and working with the mission of "Complete Client Success".

Some of there partner are IBM, MICROSOFT, REDHAT, AWS etc.

MY ROLE IN COMPANY

As I Intern as Software Engineer I have to undergo through learning process which includes 8 modules to learn that are Git, SQL , OOPs , HTML, Spring , Maven , Java and JavaScript. After which I will be allocated to advance training phase and a project.

Chapter 2: Technologies used

SQL (STRUCTURED QUERY LANGUAGE)

what is SQL

- SQL stands for Structured Query Language. It is a special purpose language designed for managingdata 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

Types of SQL Statements

1. DATA RETRIEVAL :- Retrieves data from the databaseStatement :-
   - SELECT.

2. DATA MANIPULATION LANGUAGE (DML) :- used to manage data within schema objectsStatement :-
   - 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. Changesto 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 andstructures within it.Statement :-

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

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 tablestogether.
```

• 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.

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:-

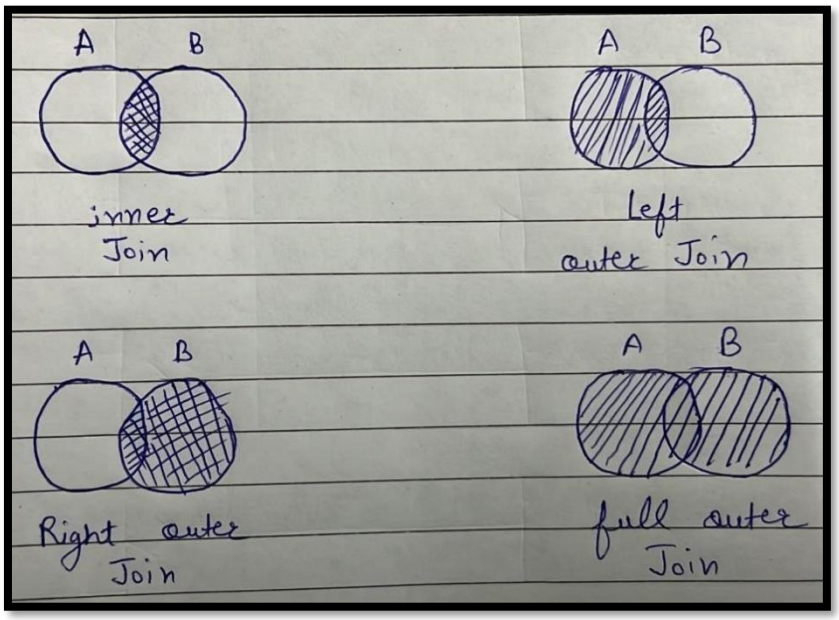


Fig 4.3.1 Types of joins

2.1.2 OOPS

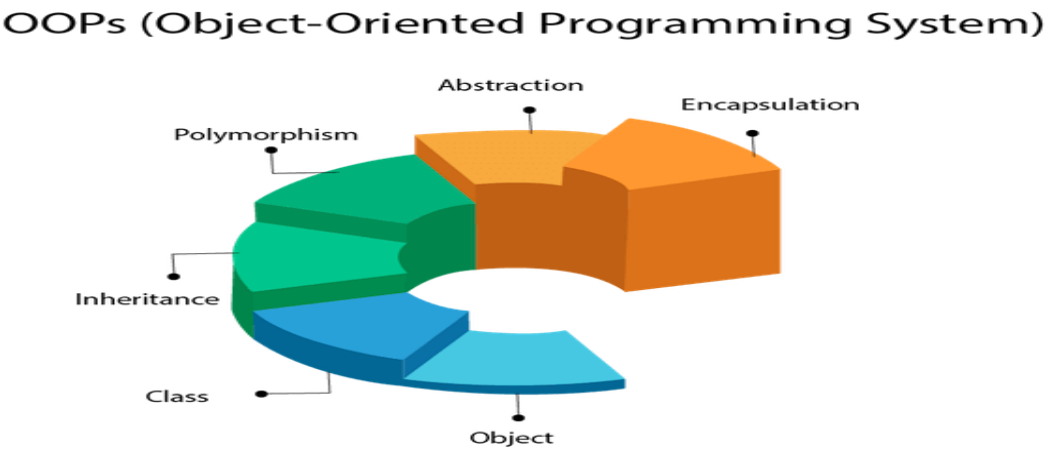
OOPs (Object-Oriented Programming System)

The meaning of Object is real-world entity such as a pen, table, computer, watch, etc. **Object-Oriented Programming** is methodology or say paradigm to design an program using classes & objects. It simplifies software development and maintenance by providing some concepts:

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

Apart from these concepts, there are some other terms which are used in Object-Oriented design:

- Coupling
- Cohesion
- Association
- Aggregation
- Composition



Object

Any entity that has a state and behavior is known as an object. For example, a chair, pen, table, keyboard, bike, etc. It can be physical or logical.

An Object can be defined as an instance of a class. An object contains an address and takes up some space in memory. Objects can communicate without knowing the details of each other's data or code. The only necessary thing is the type of message accepted and the type of response returned by the objects.

Example: A dog is an object because it has states like color, name, breed, etc. as well as behaviors like wagging the tail, barking, eating, etc.

Class

Collection of objects is called class. It is a logical entity.

A class can also be defined as a blueprint from which you can create an individual object. Class doesn't consume any space.

2.1.3 GIT and GITHUB

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.

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.

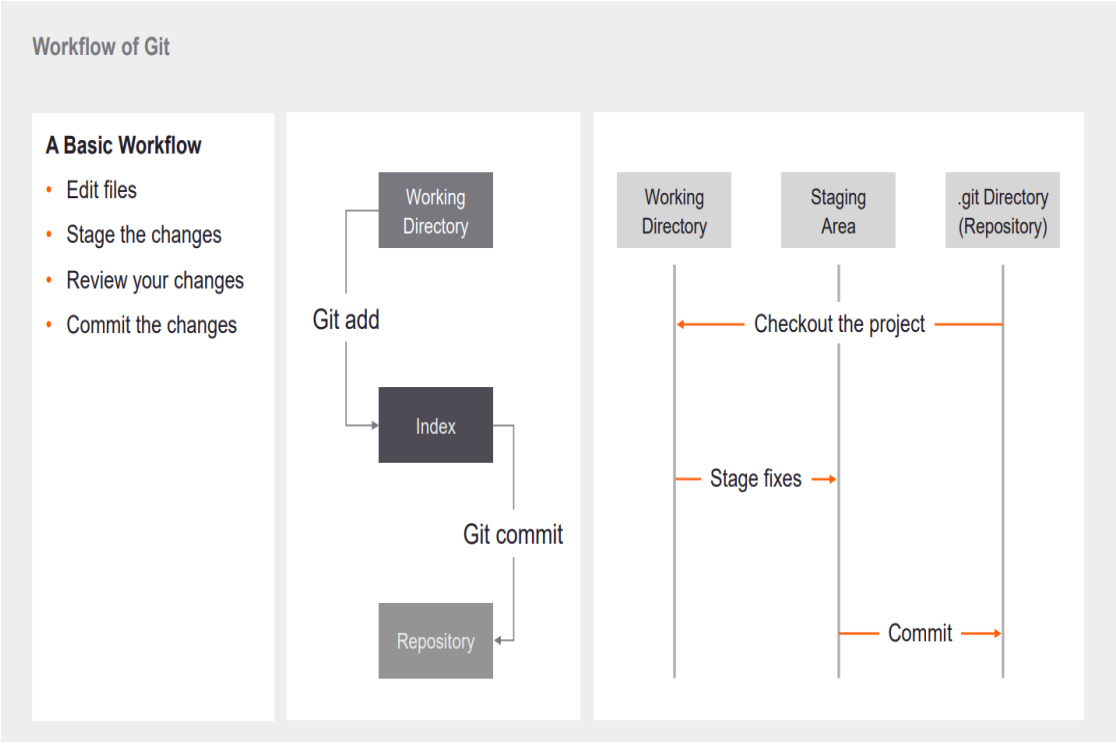
• The Three States – The Git working directory, The staging area and Repository.

• Everyone has the complete history.

• Everything is done offline.

• No central authority.

• Changes can be shared without a server.



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 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 .

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.

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 - 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 .

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

• git log - →this command display commits that include the specified file. This is an simple way to see the history of a specific file.

(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.

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.

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.

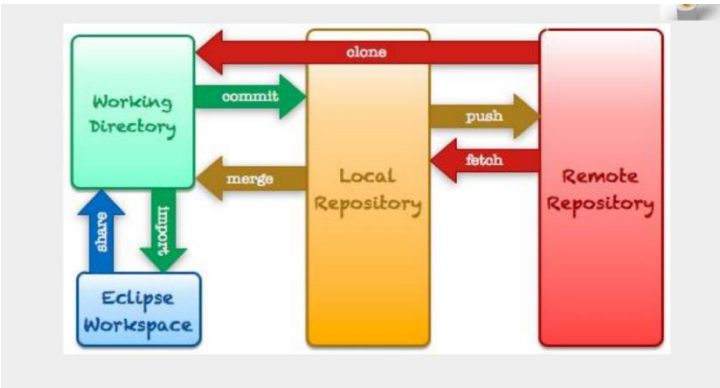
Commands :

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.



2.1.4 JAVA

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

• Architectonic Neutrality

• Object-Oriented

• Robust

• Secure

• Distributed

• Interactive and multi-threaded

• High Efficiency

• Extensible and dynamic

Object oriented program (OOPS)

What is OOP?

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

Encapsulation

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

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.

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.

.**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

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

. **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.

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.\

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**.

• If no constructor is **supplied**, a default constructor is used to create **times** of the **magnificence**. A default Constructor initializes **the example** variables with default values of the **information** type

• If **as a minimum** one constructor is **supplied** , default constructor **isn't always supplied via way of means of** JVM.

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.

Fundamental Class

Object Class

• Every **magnificence(Object)** that we create extends the **magnificence** Object **with the aid of using** default. That is the Object **magnificence** is at **the best** of the hierarchy. This **allows** to **byskip** an **item** of any **magnificence** to be **handed** as **an issue** to methods. Every **magnificence** that we create extends the **magnificence** Object **with the aid of using** default. That is the Object **magnificence** is at **the best** of the hierarchy. This **allows** to **byskip** an **item** of any **magnificence** to be **handed** as **an issue** to methods.

• Some of the methods in this class are:

Method name	Description
equals(Object ref)	returns true if both objects are equal

finalize( ) method called when an object's memory is destroyed

toString() return the string equivalent of the object name

notify( ) method to give message to a synchronized methods

notifyAll( ) method to give message to all synchronized methods

wait() suspends a thread for a while

Math Class

• Support **basic** mathematical functions.

• Can **now no longer** be instantiated.

• Round off Functions

- Exponential Functions

static double pow(double d1, double d2) :- This Is Explonential static function and return value in double.

- Trigonometric Functions

static double sin(double d) :- This Is Trignometric static function and give angle value in double.

• random function generator

static double random()

Wrapper Class

• Are used **to control** primitive values as objects.

• Are final.

• Objects of wrapper **instructions** are immutable

• static String toString(**kind** v) :- returns the string **similar to** the primitive **fee** of **kind handed** as argument.

int compareTo(WrapperType obj2)

type parseType(String s)

AutoBoxing and Autounboxing

Autoboxing

Auto Boxing is the process where the primitive data member automatically gets converted into its respective wrapper object. No need of explicit conversion by the programmer.

Integer intObj1 = new Integer(10); //boxing Integer intObj2 = 10; //auto boxing

Autounboxing

Auto unboxing is the process of automatically extracting the primitive value wrapped inside the object. No need of calling any method explicitly to fetch the primitive value.

int num1 = intObj1.intValue(); //unboxing int num2 = intObj2; //auto unboxing

Exception Handling

What is Exception Handling?

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

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

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.

Uncheked 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.

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.

2.1.5 MAVEN

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.

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

Broader pictures

• Lifecycle - broadest unit

• Phase - A stage in the lifecycle.

• Goal - A plugin goal represents a specific task

Lifecycles in maven

• default - handles your project deployment

• clean - handles project cleaning

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

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

• **package** - take the compiled code and package it in its distributable format, such as a JAR.

• **verify** - run any checks on results of integration tests to ensure quality criteria are met

• i**nstall** - install the package into the local repository, for use as a dependency in other projects locally

• **deploy** - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects

Maven Coordinates

- The Archetype plugin created a assignment(project) with a document named pom.xml.
- This is the Project Object Model (POM), a declarative description of a assignment
- Maven coordinates outline a hard and fast of identifiers which may be used to uniquely become aware of a assignment .
    - groupId
    - artifactId
    - Version (Snapshot/1.x/2.x)
    - Packaging (jar/war/ear/pom)
- Packaging – (Jar/war/ear/pom) Represent the package for the Project in which we work.
- artifactId – Represent the name of the Project .
- groupId – it is the combination of packaging and artifactid to get unique groupis.
- Version – represent the current version of the project.

Maven POM

• Stands for Project Object Model

• Describes a assignment (project)

- Name and Version

- Artifact Type

- Dependencies

- Plugins

- Profiles

• By default They uses Xml file.

Project Object Model

• Maven projects, dependencies, builds, artifacts: all **of those** are **items** to be modeled and **defined withinside the** POM **report**.

• The POM tells Maven what **type of project it is.** Also managing modifying default behaviour to bring out output from source.

• Maven project is **described via way of means of** the presence of a pom.xml

• It is a descriptive an**nouncement** of a project for Maven; **it's far** the figurative “map” that Maven **desires** to understand

• • Somewhat analogous to the **construct report** in ant.

2.2 Framework

2.2.1 SPRING

- "Spring is one of the most popular frameworks for Java enterprise edition."
- "It was developed by Rod Johnson in 2003."
- Spring framework makes the development easy of JavaEEapplication.
- Developers all over the world use Spring for developing reliable and high-quality applications.
- It is basically platform where we can find the solution to technicalproblems.
- Advantages of Spring Framework are— These frameworks have loose couplings.

  These frameworks have predefined templated that is we do notneed to write the whole code.

  It is easy to test the application here.It provides the powerful abstraction.

  It is lightweight frame work because of its POJO implementation.

In this module I basically learnt about the basics of Spring framework.

2.3 Front-End Technologies

2.3.1 HTML

2.3.2 CSS

2.3.3 JS

HTML, CSS, & JavaScript: A Tutorial

An overview:

HTML provides the *basic structure* of sites, which is enhanced and modified by other technologies like CSS and Java Script.

CSS is used to control *presentation, formatting, and layout*.

JavaScript is used to control the *behavior* of different elements.

FPR-1

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Akhil Umoriya		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	12/01/22-12/01/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work				✓	
Learning capacity/Knowledge up gradation			✓		
Performance/Quality of work			✓		
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	Learned about GIT, SQL and Concept of OOPs.				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	Jayati Munot				

Receiving Date		**Name of Faculty Mentor**		**Sign**	

FPR-2

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Akhil Umoriya		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	16/02/22-2/03/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work					✓
Learning capacity/Knowledge up gradation			✓		
Performance/Quality of work			✓		
Behaviour/Discipline/Team work			✓		
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	GIT, SQL and Concept of OOPs practice , Learning in Java core				
OVERALL GRADE (Any one)	**GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	Jayati Munot				

Receiving Date		**Name of Faculty Mentor**		**Sign**	

FPR-3

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Akhil Umoriya		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	3/03/22-15/03/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work					✓
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work			✓		
Behaviour/Discipline/Team work			✓		
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	Java core , Maven				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	Jayati Munot				

Receiving Date		**Name of Faculty Mentor**		**Sign**	

FPR-4

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Akhil Umoriya		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	15/03/22-30/03/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work					✓
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work				✓	
Behaviour/Discipline/Team work			✓		
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	Maven , javascript, html, spring boot				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	Jayati Munot				

Receiving Date		**Name of Faculty Mentor**		**Sign**	

FPR-5

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Akhil Umoriya		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	1/04/22-15/04/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work				✓	
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work				✓	
Behaviour/Discipline/Team work			✓		
Sincerity/Hard work			✓		
Comment on nature of work done/Area/Topic	javascript, html				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	*Jayati Munot*				

Receiving Date		**Name of Faculty Mentor**		**Sign**	

FPR-6

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Akhil Umoriya		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	15/04/22-30/04/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work				✓	
Learning capacity/Knowledge up gradation					✓
Performance/Quality of work				✓	
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work			✓		
Comment on nature of work done/Area/Topic	JS, HTML				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	Jayati Munot				

Receiving Date		**Name of Faculty Mentor**		**Sign**	

FPR7

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Akhil Umoriya		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	1/05/22-15/05/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work				✓	
Learning capacity/Knowledge up gradation					✓
Performance/Quality of work				✓	
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work			✓		
Comment on nature of work done/Area/Topic	JS, HTML, CSS				
OVERALL GRADE (Any one)	**GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	Jayati Munot				

Receiving Date		**Name of Faculty Mentor**		**Sign**	

CONCLUSION

The work experience I encountered during the internship allowed me to develop technical skills such as Git, SQL, HTML, JAVA and more also I have learned softskills such as Communication, leadership skills and management through the power skills sessions which were held every Friday also there were ELC sessions on alternate weeks which helped to learned and enhance our creativity level and team management skills as we were divided into teams and work together this helped us to gain better understanding and increase our capability to listen , communicate and work in more efficient way.

REFERENCES

1. Persistent material
2. Udemy Material
3. https://en.wikipedia.org/wiki/Git
4. Java OOPs Concepts - Javatpoint
5. https://online.visual-paradigm.com/diagrams/solutions/free-uml-tool
6. https://www.javatpoint.com/what-is-core-java
7. https://www.javatpoint.com/spring-mvc-model-interface
8. Spring tutorial examples eclipse (w3schools.blog)
9. https://www.javatpoint.com/maven-tutorial

Persistent

Generated Time : 26-05-22 12:08

Progress Report

A

Akhil Umoriya
akhil_umoriya@persistent.com

14
Days Worked

84h
Time Spent

96
Overall Participation

95
Overall Performance

Java Program

Batch Name : PSL_JAVA4_JAN_BATCH_02

Batch Owner : Bhagyashree Vikhe
bhagyashri_vikhe@persistent.com

Course Learning Metrics

S.No	Course Name	Skill Name	Participation	Performance
1	Persistent - SQL Queries	SQL	100	100
2	Persistent - Git For Beginners	DevOps	100	100
3	Persistent - OOPs Concepts	OOPs	100	-NA-
4	Persistent - Core Java & Junit	Java	100	100
5	Persistent - HTML	HTML	78	100
6	Persistent - JavaScript	Javascript	73	100
7	Persistent - Maven	Maven	71	-NA-
8	Persistent - Spring	Spring	64	100

Assessment Summary

F 0-69 B 70-79 A 80-89 A+ 90-100

S.No	Assessment Name	Attempted Date	Grade	Status
1	GEMS Generic Paper - Combined Test	04 Mar 2022	A+	Cleared
2	GEMS Core Java and Junit - Combined Test	17 Mar 2022	B	Cleared
3	GEMS JS_HTML_CSS Combined_Test	03 May 2022	A	Cleared
4	Spring Boot and Maven	19 May 2022	A+	Cleared
5	GEMS Powerskill	Not Attempted	-	-