

INTERNSHIP REPORT

Internship report submitted in partial fulfilment of the requirement for the
degree of Bachelor of Technology

In

COMPUTER SCIENCE ENGINEERING

By:

Sarthak Pachauri (171308)

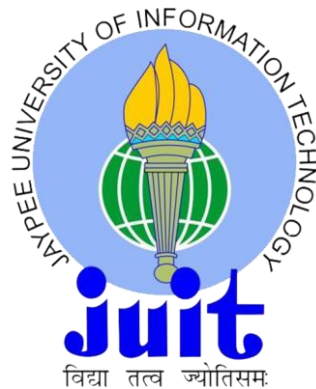
Under the supervision

Of

Mr. Surjeet Singh

Assistant Professor (Grade-II)

To



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Wanknaghat,

Solan-173234, Himachal Pradesh

PROJECT REPORT UNDERTAKING

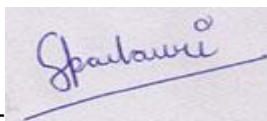
I Sarthak Pachauri Roll No-171308 Branch-Computer Science and Engineering (CSE) doing my internship with Cognizant Technology Solutions (CTS) from 6 March 2021 to 16 August 2021

As per procedure I have to submit my project report to the university related to my work that I have done during this internship.

I have compiled my project report but due to COVID-19 situation my project mentor in the company is not able to sign my project report.

So, I hereby declare that the project report is fully designed/developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to TNP Cell whenever COVID-19 situation gets normal.

Signature-



Name-Sarthak Pachauri

Roll No-171308

Date-21-05-2021

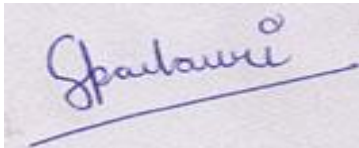
Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled "Internship Report" in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from March 2021 to May 2021 under the supervision of Mr. Surjeet Singh (JUIT) and Mr. Abhijit Joshi (**Trainer at Cognizant**).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Sarthak Pachauri, 171308



This is to verify that the above statement made by the candidate is true to the best of my knowledge.



Mr. Surjeet Singh
Assistant Professor (Grade-II)

Mr. Abhijit Joshi
Trainer at Cognizant

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to Mr. Abhijit Joshi, Trainer at Cognizant for constantly monitoring and guiding us to the right path in terms of the various skills we learned during these weeks. He constantly helped us in our research and the learning wouldn't be possible without his constant support. I would like to record my sincere appreciation and gratitude towards all the officials and employees of cognizant, without whose kind assistance, my internship program would not have succeeded. The facts and other vital information provided by them have contributed towards making this report as comprehensive as possible. I am indeed, thankful to them. I would also like to express my gratitude to my college project supervisor Mr. Surjeet Singh for his valuable guidance and help in completing this project report.

TABLE OF CONTENT

1. CHAPTER-1: ORGANIZATION	(7)
Background	(7)
Mission	(7)
Vision and Values.....	(8)
Objective	(8)
2. CHAPTER-2: PROGRAM STRUCTURE.....	(9)
Program Structure.....	(9)
Functional Testing	(10)
Java Programming and Fundamentals.....	(15)
Web UI and Datasource	(28)
3. CHAPTER-3: PERFORMANCE ANALYSIS.....	(45)
4. CHAPTER-4: CONCLUSION	(46)
Conclusion	(46)

References

LIST OF FIGURES

Figure No.	Description	Page No.
FIG-1	Program Structure	9
FIG-2	SDLC Model	11
FIG-3	Verification and Validation	11
FIG-4	Functional Testing	12
FIG-5	Automation Testing	13
FIG-6	Traditional and Agile SDLC	13
FIG-7	Test Case Template	14
FIG-8	JVM	14
FIG-9	Non-Primitive Data Types	15
FIG-10	Variable	15
FIG-11	Class and Object	20
FIG-12	Abstract class and Interface	22
FIG-13	JDBC Architecture	26
FIG-14	HTML Page Structure	28
FIG-15	Sample HTML Code	30
FIG-16	XML Encoding	30
FIG-17	JSON Sample	30
FIG-18	JavaScript Sample	32
FIG-19	CSS	33
FIG-20	CSS Sample	34
FIG-21	Grid System	35
FIG-22	Bootstrap Sample	35

LIST OF TABLES

Table No.	Description	Page No.
Table-1	Primitive Data Types	15
Table-2	Basic HTML Tags	17
Table-3	Input and Form Tags	17
Table-4	Table Tags	19
Table-5	Entities	20
Table-6	Different Types of Selector	40

CHAPTER-1: ORGANIZATION

(1.1) Background

Cognizant is a top IT company in India, and major IT company in US. Cognizant employees are around 3 lakhs and recruits around 20 thousand fresh people every year from India. Cognizant also hires from different country across the globe.

Cognizant offers various role in the company like develop, Designer, Tester and Manager in the company, but, before becoming an associate every person should complete the internship period and after the intern period there is one year of probation period in the company.

The internship period varies and depends on the roles, which the intern gets, like someone who got developer profile, for them internship period will be of around 4-5 months and for the quality insurance, it might vary from 5-6 month. The domain allocation is random in the cognizant for the interns, but sometime it depends on the assimilation test also, the person who got higher marks in assimilation test, will have higher chances to get better profile or domain and it also depends on the first come first serve basis.

(1.2) Mission

The only objective of Cognizant is to work with customers to strengthen organisations via technology and process innovation, deep industry experience, and a global resources.

(1.3) Vision

Cognizant Company's vision statement, which describes the company's long-term strategy, identifies the many developing demands of the market, different consumer habits, and current market conditions that will try to push the information economy.

(1.4) Values

Company values are as follows:

1. First and foremost, people are important to a company's success. When we respect everyone, we interact with, we will make and keep strong connections with our customers, coworkers, referral sources, suppliers, and the community.
2. Goal is to generate long-term customer loyalty by establishing trust, trustworthiness, and honesty in our work and providing customers with impartial advice.
3. Holding committed to principles of excellence, honesty, respect, fairness, and professionalism at all times throughout our organisation.

(1.5) Objectives

The objectives of Cognizant are:

1. Start with a point of view.
2. Seek data, build knowledge.
3. Always strive, never settle.
4. Work as one.
5. Create conditions for everyone to thrive.
6. Do the right thing, the right way

CHAPTER-2: PROGRAM STRUCTURE

(2.1) Program Structure

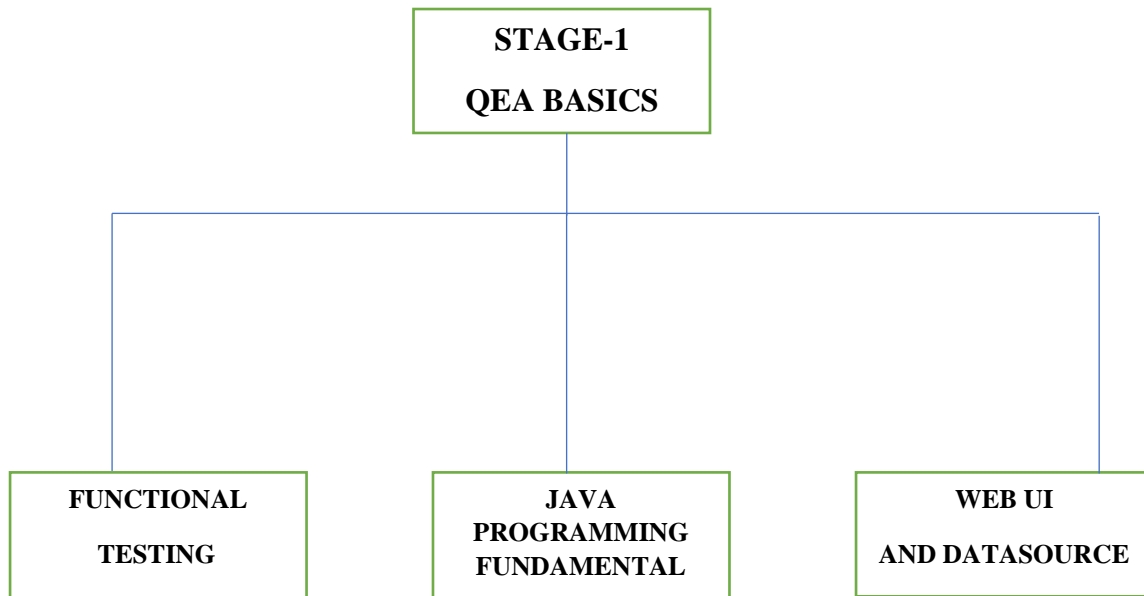


FIG-1: Program Structure

Assessment:

1. We like hands-on activities and quizzes to allow us to demonstrate our knowledge of the material.
2. Module/Skill Code Challenges are coding problems to test and assess programming abilities.
3. An Integrated Capability Test (ICT) is a case study of all the concepts learned in a single module during an integrated learning opportunity.

(2.2) Functional Testing

Week-1

(2.2.1) Testing

A test is a method of determining whether or not a system or a component meets the defined criteria. Verifying is doing a system run to find any deficiencies or mistakes in contradiction to the specifications. Testing may be characterised as it is performed in multiple formats across SDLC:

1. Analysis and verifying of criteria also are considered testing during the requirement collection process.
2. In design review, you look at the design to try to enhance it.
3. On finishing of the code, testing conducted by a developer is also known as testing.

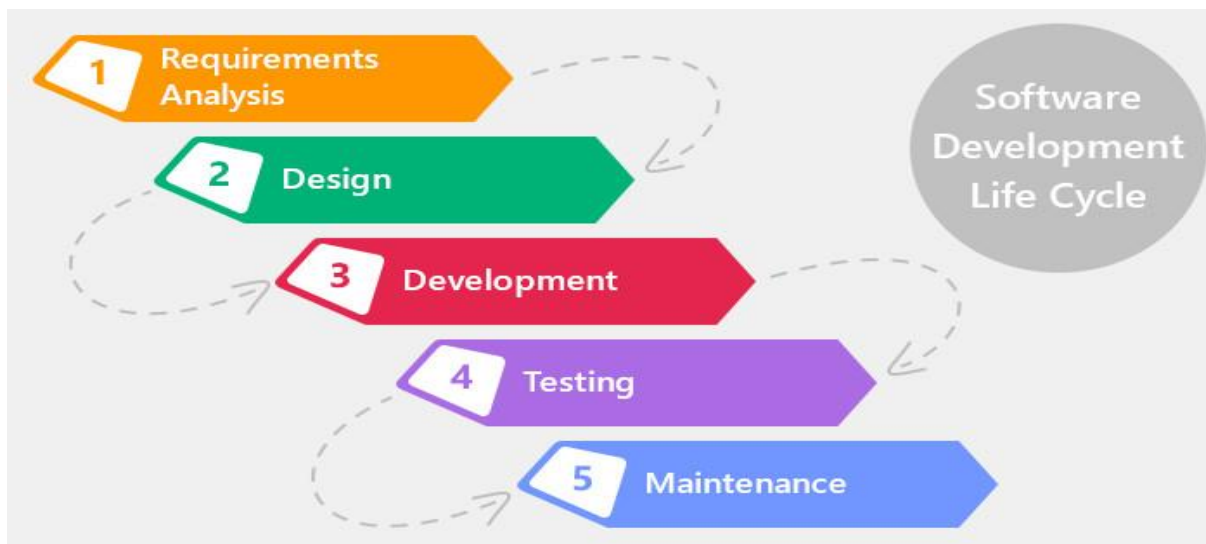


FIG-2: SDLC Model

Testing should strive to accomplish the following three basic goals: verification and validation.

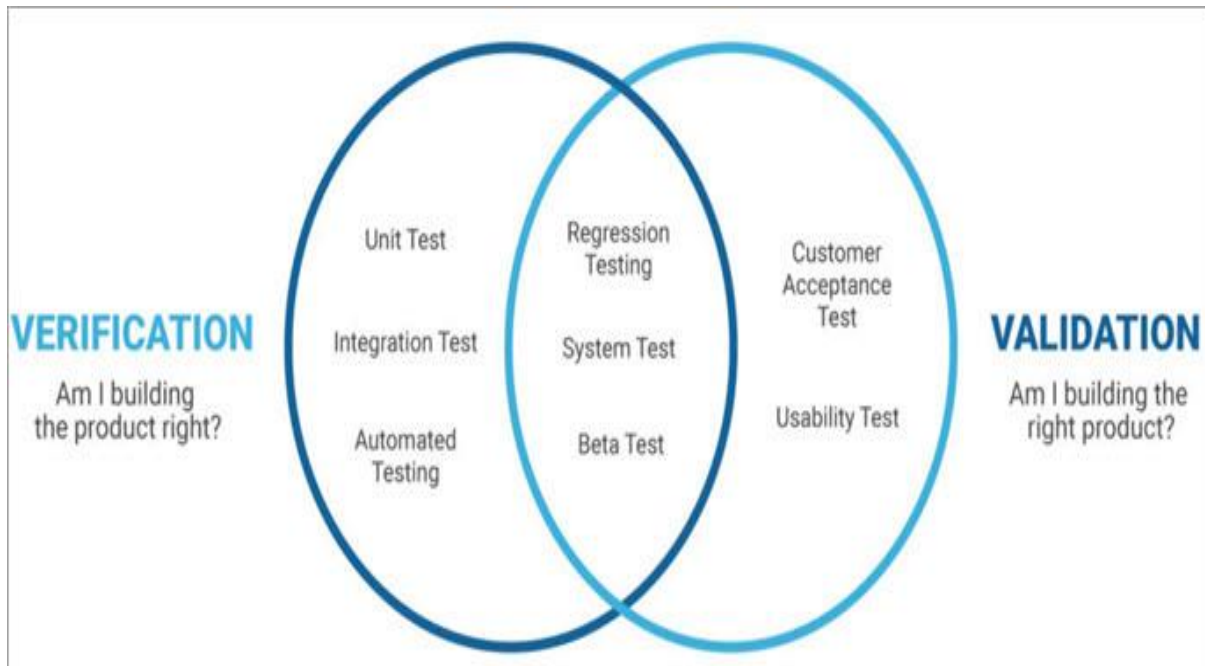


FIG-3: Verification and Validation

(2.2.2) Functional Testing

Function testing is a sort of application testing that is conducted against the company's business needs. The sort of testing is known as a black box test. It involves the whole integration system to assess the platform conformity with its set requirements. This form of testing is laid out in the functional specification document.

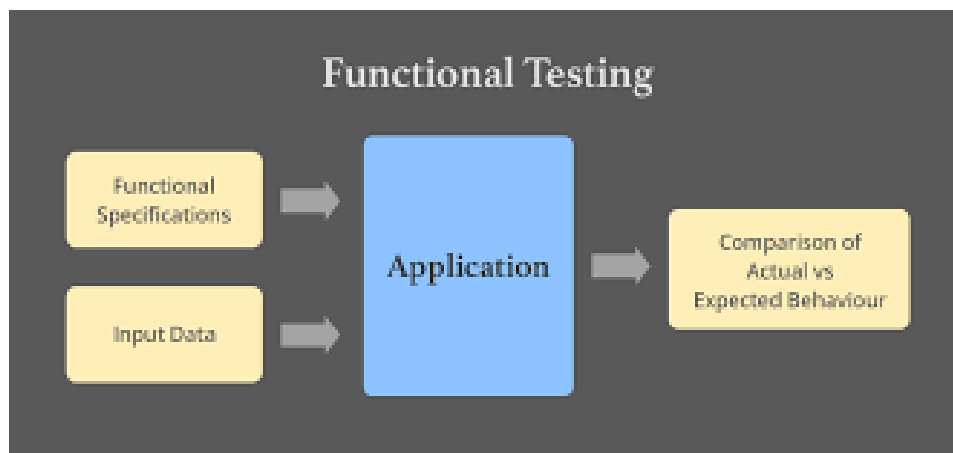


FIG-4: Functional Testing

(2.2.3) Types of Testing

1. Manual Testing- In manual testing, you do manual testing, which means you use no automated tool or script. This form of test is done by the tester becoming an end-user and executing the programme to search for any unexpected or critical behaviour.
2. Automation Testing- Automation testing, often known as test automation, is a method through which a tester utilises scripts and other tools to perform a variety of tests on a product.

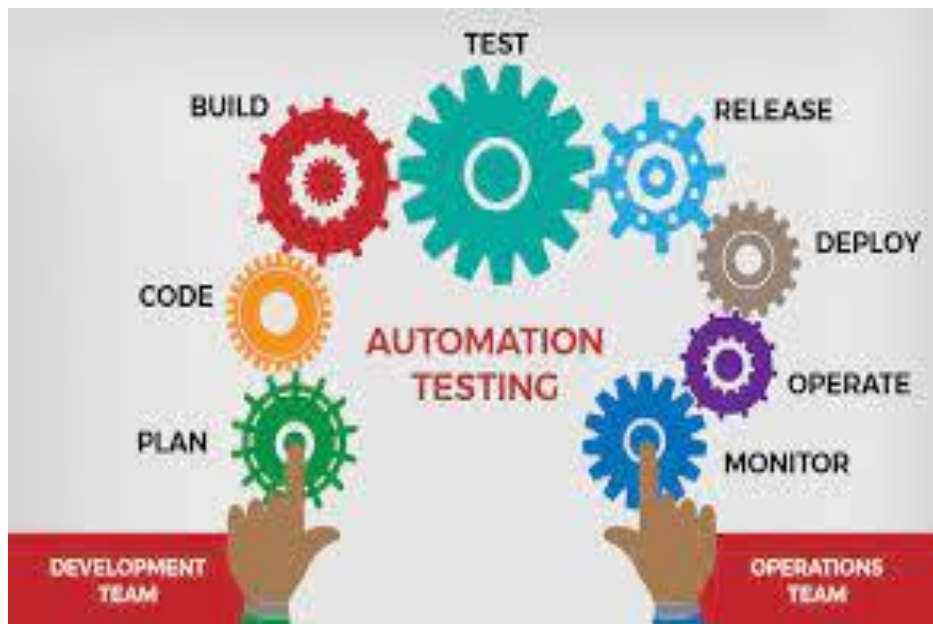
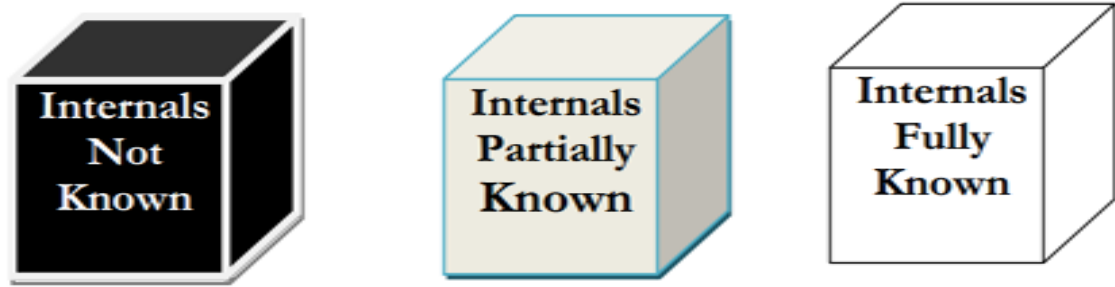


FIG-5: Automation Testing

(2.2.4) Testing methods



Comparison between the Three Testing Types

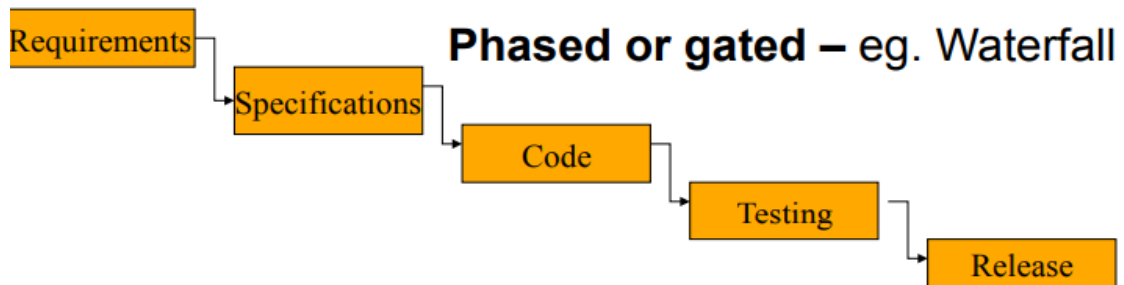
	Black Box Testing	Grey Box Testing	White Box Testing
1.	The Internal Workings of an application are not required to be known	Somewhat knowledge of the internal workings are known	Tester has full knowledge of the Internal workings of the application
2.	Also known as closed box testing, data driven testing and functional testing	Another term for grey box testing is translucent testing as the tester has limited knowledge of the insides of the application	Also known as clear box testing, structural testing or code based testing
3.	Performed by end users and also by testers and developers	Performed by end users and also by testers and developers	Normally done by testers and developers
4.	-Testing is based on external expectations -Internal behavior of the application is unknown	Testing is done on the basis of high level database diagrams and data flow diagrams	Internal workings are fully known and the tester can design test data accordingly
5.	This is the least time consuming and exhaustive	Partly time consuming and exhaustive	The most exhaustive and time consuming type of testing
6.	Not suited to algorithm testing	Not suited to algorithm testing	Suited for algorithm testing
7.	This can only be done by trial and error method	Data domains and Internal boundaries can be tested, if known	Data domains and Internal boundaries can be better tested

FIG-6: Types of Testing.

(2.2.6) Agile Testing

At order to promote the introduction of testers at the conclusion of the development process, the notion of including them in the beginning emerged, which paved the way for the introduction of agile testing. The start of the project represents the start of the agile testing era.

Traditional vs. Agile SDLC



Agile: iterative and incremental

- Each story is expanded, coded and tested
- Possible release after each iteration

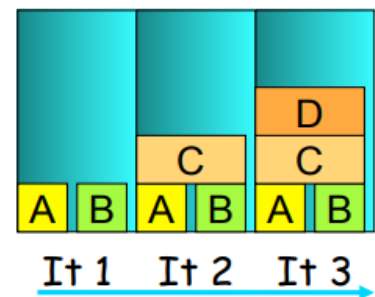


FIG-7: Traditional Vs Agile SDLC

(2.2.7) Test Case Writing

Test Case-A test case would be a collection of variables or situations that an engineer will use to find out if a product under test is accurate and conforms to specifications. One must include a wide range of fields in a test case.-

1. Test Case ID.
2. Module to be tested.
3. Assumptions-If any
4. Test Data
5. Test Steps
6. Expected results.
7. Result-Pass or Fail
8. Comments

Project Name:						
Test Case Template						
Test Case ID: Fun_10			Test Designed by: <Name>			
Test Priority (Low/Medium/High): Med			Test Designed date: <Date>			
Module Name: Google login screen			Test Executed by: <Name>			
Test Title: Verify login with valid username and password			Test Execution date: <Date>			
Description: Test the Google login page						
Pre-conditions: User has valid username and password.						
Dependencies:						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to login page	User= sample@gmail.com	User should be able to login	User is navigated to	Pass	
2	Provide valid username	Password: 1234		dashboard with successful		
3	Provide valid password			login		
4	Click on Login button					
Post-conditions:						
User is validated with database and successfully login to account. The account session details are logged in database.						

FIG-8: Test Case Template

(2.3) Java Programming Fundamentals

Week-2

(2.3.1) Java Overview

Java is really a strong, class-based, object-oriented programming language with the objective of minimizing implementation dependencies. Because compiled Java code can execute on any systems that accept Java, the requirement for recompilation is eliminated.

Principles

1. Object-oriented, straightforward, and familiar must be in the final build.
2. To be secure and durable, it must be sturdy.
3. Architecture-neutral and portable is a requirement.
4. It must run quickly.
5. The creation of a dynamic, interpreted, and interwoven meaning must be understood.

(2.3.2) JVM

A JVM is a run-time engine that can execute Java programmes. The JVM, the Java Virtual Machine, initiates the call to the main function inside the Java code. JVM is a component of the JRE (Java Runtime Environment). In contrast to Perl, Java applications are known as (WORA-Write Once Run Anywhere). When this is said, it implies that programmers may build Java source code on one machine and have confidence that it will operate on any other Java-capable machine without any modification. Due to the existence of the JVM, this is all feasible.

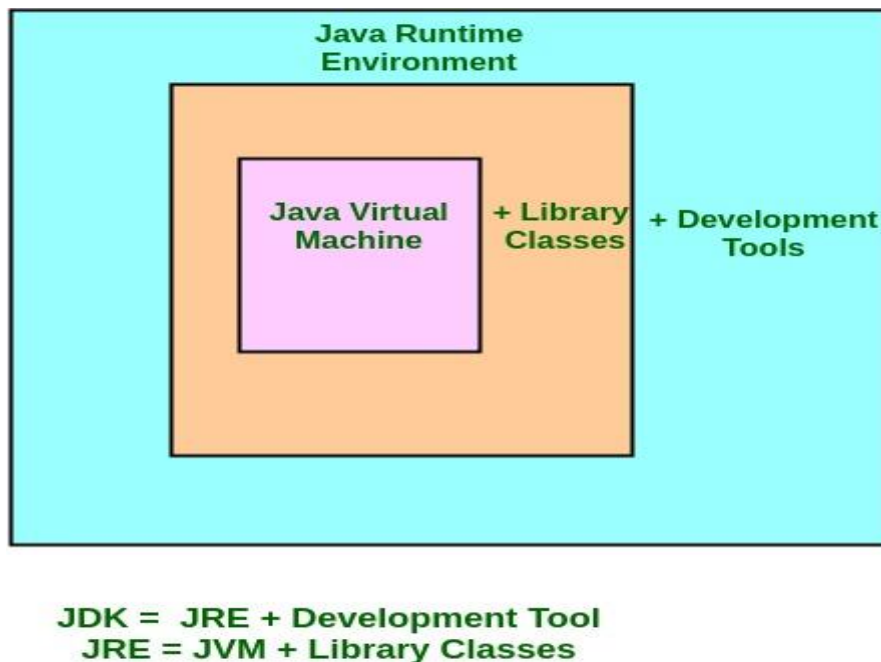


FIG-9: JVM

JAVA DEVELOPMENT KIT

The Java Development Kit (JDK) is a Programming language software development environment. The Java Development Kit (JDK) is composed of the Java Runtime Environment (JRE), an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), and other tools required in Java development.

JAVA RUNTIME ENVIRONMENT

Java Runtime Environment (JRE) is another name for Java RTE. The Java Runtime Environment (JRE) includes the Java Virtual Machine (JVM), core classes, and auxiliary files, which provide the necessary for running a Java application.

(2.3.3) Java Identifiers

The aim of identifiers in programming languages is to allow for identification. An identifier in Java may be any of the following: a class name, a method name, a variable name, or a label. For instance:

```
public class Test  
  
{  
  
    public static void main (String [] args)  
  
    {  
  
        int a = 20;  
  
    }  
  
}
```

Within the java code shown above, we have five distinct identifiers:

Test: class name.

main: method name.

String: predefined class name.

args: variable name.

a: variable name.

(2.3.4) Data Types

Since every kind of data is declared explicitly element of a programming language in Java, it is a statically typed language, as well as a strongly typed language. When it comes to data, Java has two distinct categories:

1. Primitive Data Type: such as Boolean, char, int, short, byte, long, float, and double
2. Non-Primitive Data Type or Object Data type: such as String, Array, etc.

Primitive Data Types

Table-1: Primitive Data Types

TYPE	DESCRIPTION	DEFAULT	SIZE	EXAMPLE LITERALS	RANGE OF VALUES
boolean	true or false	false	1 bit	true, false	true, false
byte	twos complement integer	0	8 bits	(none)	-128 to 127
char	unicode character	\u0000	16 bits	'a', '\u0041', '\t01', '\l', '\', '\n', '\b'	character representation of ASCII values 0 to 255
short	twos complement integer	0	16 bits	(none)	-32,768 to 32,767
int	twos complement integer	0	32 bits	-2, -1, 0, 1, 2	-2,147,483,648 to 2,147,483,647
long	twos complement integer	0	64 bits	-2L, -1L, 0L, 1L, 2L	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	IEEE 754 floating point	0.0	32 bits	1.23e100f, -1.23e-100f, .3f, 3.14F	upto 7 decimal digits
double	IEEE 754 floating point	0.0	64 bits	1.23456e300d, -1.23456e-300d, 1e1d	upto 16 decimal digits

Non-Primitive Data Type or Reference Data Types- Because the reference data types wouldn't keep a variable value immediately in memory, any reference data types will now have a memory address that refers to the variable value. They are strings, objects, arrays, etc.

Non-Primitive Data Types

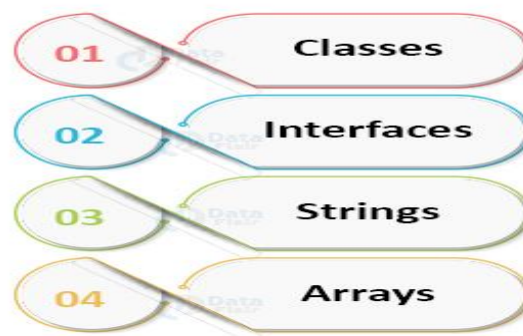


FIG-10: Non-Primitive Data Types

(2.3.5) Variables

A variable is indeed a memory region with a name. A storage unit is one of the fundamental building blocks of a programme. Variables have the ability to modify their value while a programme is running.

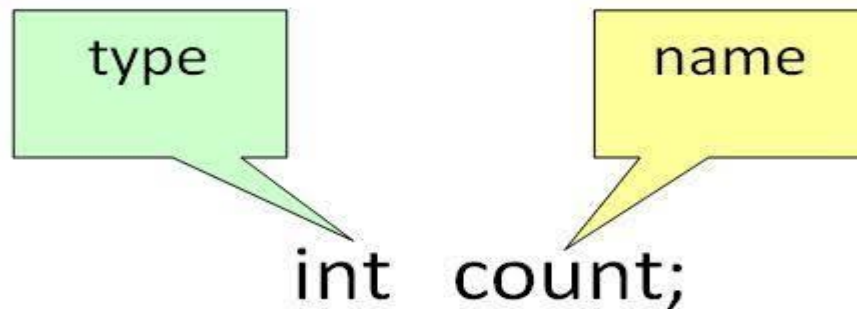


FIG-11: Variable

(2.3.6) Type Conversion

Assigning a data from one data type to the other may not be possible. To facilitate automated type conversion, Java can automatically convert the data types for you if they are equivalent. Otherwise, you must manually convert or cast the values

(2.3.7) Loops

If a certain condition is met, the instructions/functions are repeated.

While loop: A while loop would be a control flow expression that repeats code depending on a Boolean condition that holds true.

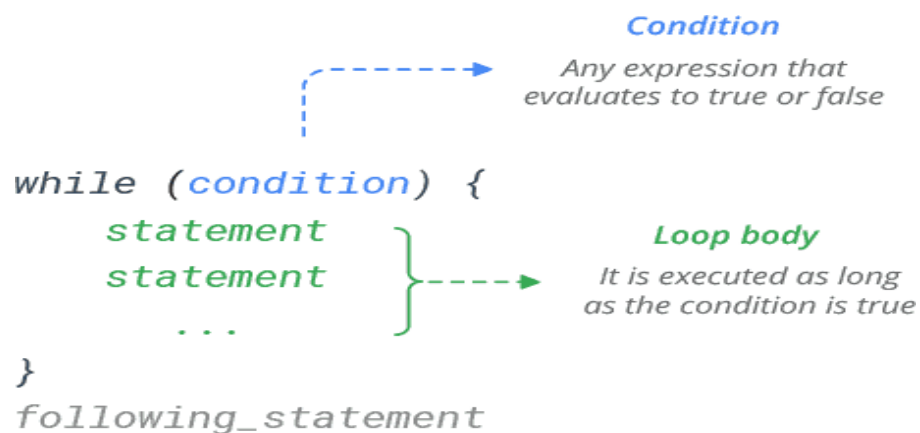


FIG-12: While loop

For Loop: The for loop is an easy method to represent the structure of a loop. On a for loop, you may have just one initialization, one condition, and one increment or decrement statement in one line, which is helpful when you are trying to analyse the structure of your loops.

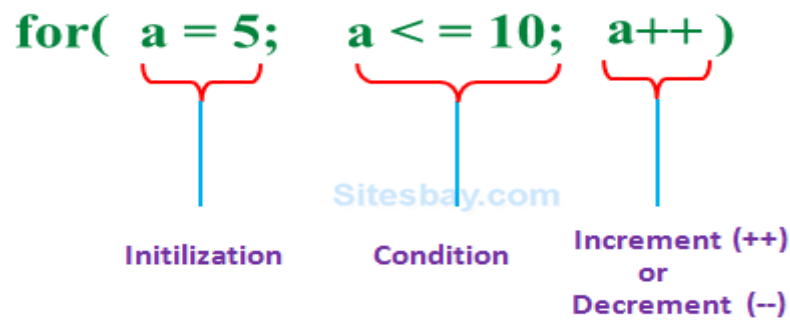


FIG-13: For loop

do while: An example of Exit Control is the do while loop.

```

Syntax:
do
{
    ...
    block of statements;
    ...
} while( condition ) ;

```

FIG-14: do while loop

(2.3.8) Decision Making.

1. if-else: An if statement on its own would perform a sequence of statements whenever a condition evaluates to true, and otherwise will not.

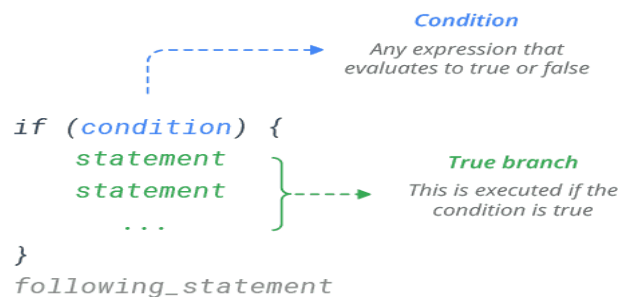


FIG-15: if-else

2. Nested-if: The if expression that would be the subject of someone else if or else is known as a nested if. If statements may be nested, therefore if statements may be nested as well.

```

if(Condition/Expression)
{
    if(Condition/Expression)
    {
        Statement(s);
    }
    else
    {
        Statement(s);
    }
}
else
{
    if(Condition/Expression)
    {
        Statement(s);
    }
    else
    {
        Statement(s);
    }
}
}

```

FIG-16: Nested if-else

3. Switch-Case -A multiway branching statement is known as a switch statement. By providing a straightforward mechanism to assign execution to various sections of script up to the value of an expression, it eliminates the confusion caused by switch statements.

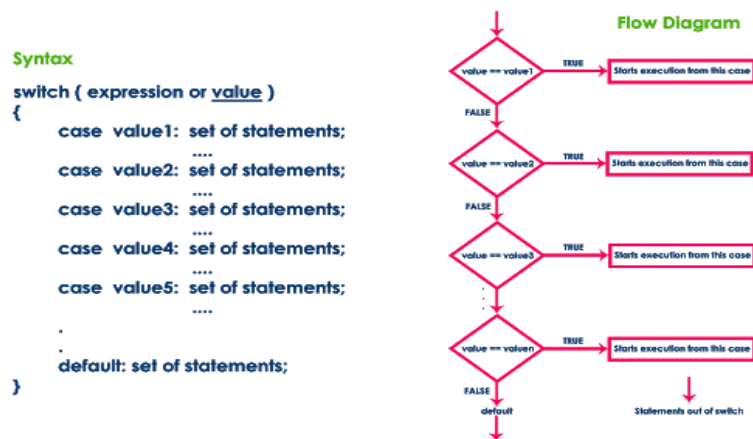


FIG-17: Switch Case

4. Jump- break, continue, and return are all supported by Java's jump statement. Control is transferred to other parts of the programme in these three phrases.

Break-A frequent usage of break is to quit a loop or to go to a different section of code inside a loop.

Continue-If forcing an initial execution of a loop may be helpful, do so. In other words, when you execute the for loop in this instance, you should continue executing the loop, but not process the rest of the script in the loop body this time around.

Return-With the return statement, you may specify that a method should exit. For instance, if a method causes a control process to transferring back to the caller, it means the method will cause the called method to return.

(2.3.9) Static Variable

In addition to the static variables described above, there are also static class variables which are shared by all the objects of either the class and which may be accessed using the class name.

(2.3.10) Non-Static Variable

A non-static variable is sometimes termed an instance variable, since every instance of a class is a separate variable from the static members of the class.

(2.3.11) Classes and Objects

Class-A class is a blueprint or template for creating things from which users derive user-defined capabilities. All things of one type have the same attributes or methods. Class declarations may comprise the following elements in this order::

1. Modifiers
2. Class name
3. Superclass (if any)
4. Interfaces (if any)
5. Body: The class body surrounded by braces, { }.

Objects are fundamental in object-oriented programming because they correspond to physical things in the actual world. Let's use an example: A shopping system that utilises the internet may feature such terms as "shopping cart", "customer", and "product".

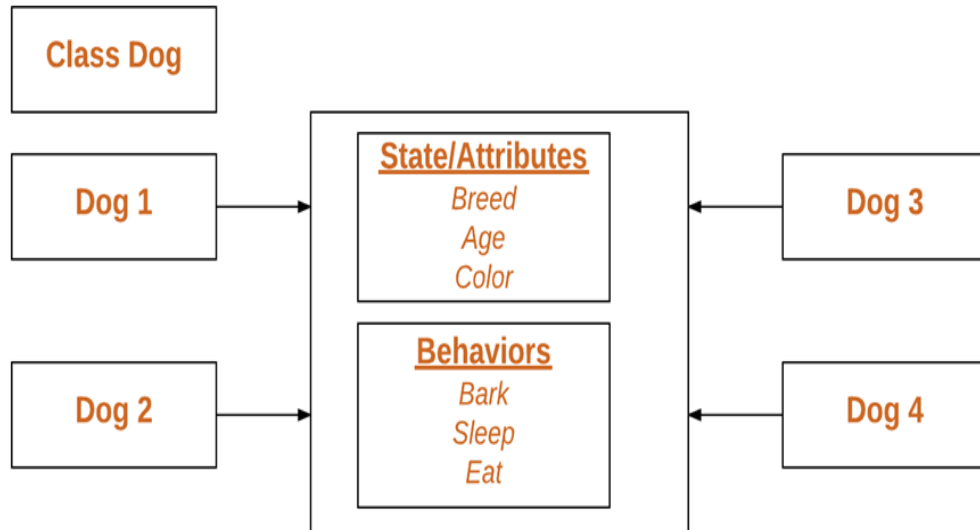


FIG-18: Class and Objects

(2.3.12) Access Modifiers

Class, function Object() { [native code] }, variable, function, and data member scope may be restricted using Java's access modifiers. Four different kinds of access modifiers, with the suffix "Default", "Private", "Protected", and "Public" are available in Java.

(2.3.13) Packages

Classes, sub packages, and interfaces may be safely and accurately packed within Java packages.

1. Name clashes are prevented through packages.
2. makes class, interface, enumeration, and annotation search and use simpler.
3. Making regulated access available
4. Packages encapsulate data (or data-hiding)

.Built-in Packages-java.lang,,java.io,java.util,java.applet,java.awt and java.net:.

(2.3.14) Inheritance

The way in which one class may inherit the characteristics (fields and methods) with another class is found in Java's implementation of a class inheritance.

Types of Inheritance in Java:

1. Single Inheritance
2. Multilevel Inheritance
3. Hierarchical Inheritance
4. Multiple Inheritance (Through Interfaces)
5. Hybrid Inheritance (Through Interfaces)

(2.3.15) Constructors

The state of an object is initialised using constructors. A function `Object () { [native code] }`, like other methods, also includes statements that are performed when the object is first created.

The different kinds of constructors include:

1. A function `Object() { [native code] }` with no parameter is called a "no-argument function `Object() { [native code] }`."
2. Parameterized function `Object() { [native code] }` is also known as a function `Object() { [native code] }` with parameters.

(2.3.16) Interfaces

Methods and variables in an interface must be specified as abstract by default (with method signature, no body). An interface just states how that class must do; it does not dictate how it does it. it is the class's blue print.

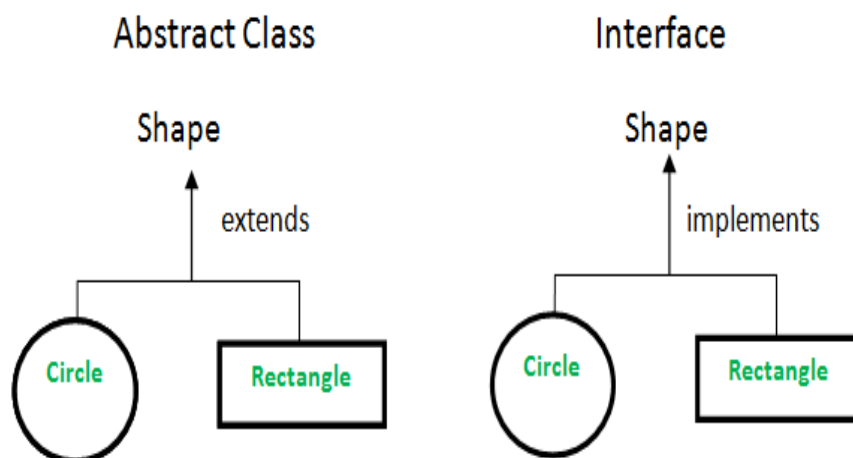


FIG-19: Abstract class and Interface.

(2.3.17) Abstraction

Data Abstraction is indeed the feature that gives you the ability to present just the necessary information to the user. Unimportant or unimportant factors are not made apparent to the user. When looking at cars, people see them as vehicles rather than separate components.

Abstract classes and methods

An abstract class is created having the abstract keyword defined inside it.

An abstract method is a method which is explicitly marked as such, yet has no implementation.

(2.3.18) Encapsulation

To encapsulate data is to enclose it inside a single unit. The means through which code and data are connected is the mechanism. Another benefit of encapsulation is

1. Data hiding.
2. It enhances flexibility.
3. Reuse
4. The testing of code is simple.

(2.3.19) Runtime Polymorphism

At the moment the call happens, Java uses the property of the object being referenced to decide which version (superclass/subclasses) of that function is to be performed. Thus, runtime polymorphism is the process by which a decision is made at runtime and this occurrence is termed runtime polymorphism.

Overloading in Java

Some methods have almost the same identity, but have various signatures, based on number of input arguments, whether the input parameters are of a certain type, or both.

Overriding in Java

Subclassing and overriding are two key concepts in object-oriented programming.

Week-3

(2.3.20) Collections

The collection of a items is made up of any set of independent things that are displayed as a singular body. The “Collection Framework” is an additional framework in Java that houses all of the collection classes & their associated interfaces in Java SE 1.2.

(2.3.21) Iterators

In the Collection Framework in Java, iterators are used to obtain items one at a time. Three iterators are involved.

1. Enumeration-To be able to get the items of older collections (Vector, Hash Table), one must utilise an enumeration.
2. Iterator- it may be applied to any Collection object; it is universal since it can be used with iterators. Because we use the Iterator, we can both read and delete data using it.
3. ListIterator-It can only be used on collection types like ArrayList, LinkedList, and the like that implement List.

(2.3.22) ArrayList

Arrays may be found in the java.util package, which belongs to the collection framework. It is Java's dynamic array.

(2.3.23) HashMap

Since Java 1.2, HashMap<K, V> is a component of Java's collection. It implements this same Map interface of Java's Map interface. The keys (or values) are organised into pairs (known as key-value pairs), and the keys may be accessed by an index that is of a different kind (e.g., an Integer).

(2.3.24) Sets

The java.util.Set interface extends the java.collections.Collection interface, offering an unsorted collection of objects that do not permit duplicate values. designated Sets in Java Collections.

1. HashSet
2. Enum Set

(2.3.25) Exception Handling

Exception was an unexpected or unwelcome occurrence that happens during the running of a programme. For example, a faulty router is an exception that may happen in a computer network. Exception handling is a method to manage runtime failures, such as the `ClassNotFoundException`, `IOException`, or `SQLException`. Exception handling's fundamental benefit is to preserve the usual operation of the application..

(2.3.26) File Handling

Java's `java.io` package provides you with practically every I/O class you'll need to use in Java. The source and destination of these many streams represents an input and an output. In the `java.io` package, you may use streams to store many kinds of data, including simple values, objects, localised characters, and more.

A stream is a series of data. As I learned to my regret, there are two sorts of streams:

Source – The source to which you will be reading data.

An Output Stream is needed to output data to a location. The 8-bit bytes that are read and written using Java bytes streams are utilised for input and output. A stream may be described as a succession of data that you may read and write. Input Stream: Used to read from an origin and to read from a destination.

Week-4

(2.3.27) JDBC

It is known as Java Database Connectivity (JDBC) in Java. JDBC is really a Java API that allows you to establish a connection and run a query on a database. The ability to use JavaSE is an element of the Java SE platform (Java Standard Edition). The JDBC API connects to the database via JDBC drivers. The four driver types are

1. JDBC-ODBC Bridge Driver,
2. Native Driver,
3. Network Protocol Driver, and
4. Thin Driver

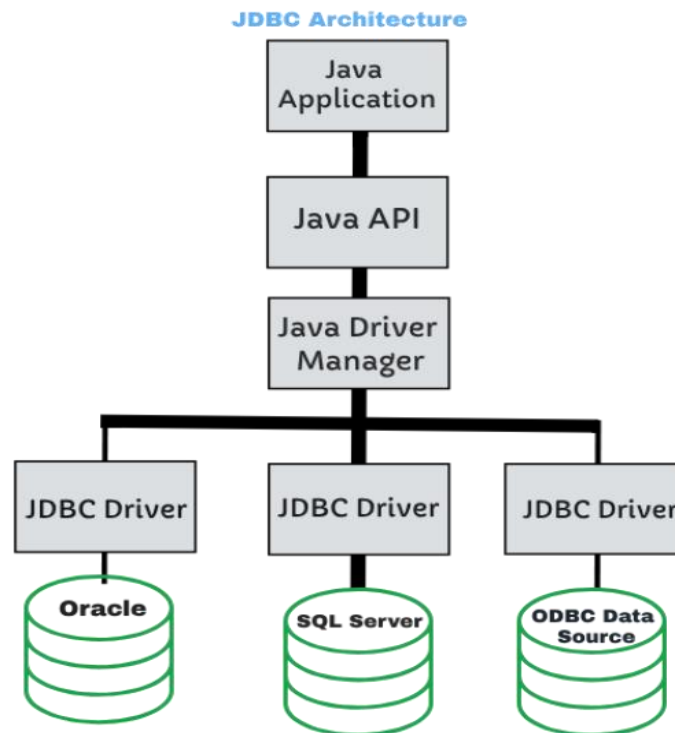


FIG-20: JDBC Architecture

We may use JDBC APIs to connect to any relational database and get tabular data contained in it. The JDBC API enables us to save, retrieve, update, and remove information from the database. It is similar to the functionality given by Microsoft's ODBC.

(2.4) Web UI And Datasource

Week-5

(2.4.1) HTML

A standard markup language for online publications, the Hypertext Markup Language, or HTML, is known as HTML. Cascading Style Sheets (CSS) and programming languages like as JavaScript may be used to help. HTML documents are served by the web server or local storage, and the browser renders the Html files creating multimedia web pages.

1. HTML is an abbreviation for the term Hyper Text Markup Language.
2. The HTML markup language is the standard means of constructing Web pages.

3. The framework of such a Web page is described by HTML.
4. HTML is made up of many components.
5. HTML components guide the browser in displaying the material



FIG-21:HTML Page Structure

Table-2: Basic HTML Tags

Tag	Description
<u><!DOCTYPE></u>	Defines the document type
<u><html></u>	Defines an HTML document
<u><head></u>	Contains metadata/information for the document
<u><title></u>	Defines a title for the document
<u><body></u>	Defines the document's body
<u><h1> to <h6></u>	Defines HTML headings
<u><p></u>	Defines a paragraph
<u>
</u>	Inserts a single line break
<u><hr></u>	Defines a thematic change in the content
<u><!--...--></u>	Defines a comment

Table-3: Input and Form Tags

Tag	Description
<code><form></code>	Defines an HTML form for user input
<code><input></code>	Defines an input control
<code><textarea></code>	Defines a multiline input control (text area)
<code><button></code>	Defines a clickable button
<code><select></code>	Defines a drop-down list
<code><optgroup></code>	Defines a group of related options in a drop-down list
<code><option></code>	Defines an option in a drop-down list
<code><label></code>	Defines a label for an <code><input></code> element
<code><fieldset></code>	Groups related elements in a form
<code><legend></code>	Defines a caption for a <code><fieldset></code> element
<code><datalist></code>	Specifies a list of pre-defined options for input controls
<code><output></code>	Defines the result of a calculation

Table-4: Table Tags

Tag	Description
<code><table></code>	Defines a table
<code><caption></code>	Defines a table caption
<code><th></code>	Defines a header cell in a table
<code><tr></code>	Defines a row in a table
<code><td></code>	Defines a cell in a table
<code><thead></code>	Groups the header content in a table
<code><tbody></code>	Groups the body content in a table
<code><tfoot></code>	Groups the footer content in a table
<code><col></code>	Specifies column properties for each column within a <code><colgroup></code> element
<code><colgroup></code>	Specifies a group of one or more columns in a table for formatting

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>TravelClient</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-e
```

FIG-22: Sample of HTML code

(2.4.2) XML

Like HTML, but without specified tags to utilise, XML is a mark - up language that is quite similar to HTML. You, however, establish your own tags with your unique requirements in mind.

```
<?xml version="1.0" encoding="UTF-8"?>
```

A small business has one department. That department has one employee. The employee details is given in the table below.

Parent Element is "Department"

Employee

empid	name	salary	email	phoneno
1001	Tom	20000	tom@gmail.com	9874563210


```

<Department>
  <Employee>
    <empid>1001</empid>
    <name>Tom</name>
    <salary>20000</salary>
    <email>tom@gmail.com</email>
    <phoneno>9874563210</phoneno>
  </Employee>
</Department>

```

FIG-23: Basic XML Document

Table-5: Entities

Entity	Character	Description
<	<	Less than sign
>	>	Greater than sign
&	&	Ampersand
"	"	One double-quotation mark
'	'	One apostrophe (or single-quotation mark)

(2.4.3) JSON

JSON is a common text-based format used to express structured data whose format is defined using JavaScript object syntax. It is often used in online applications for conveying data.

```
{
  "squadName": "Super hero squad",
  "homeTown": "Metro City",
  "formed": 2016,
  "secretBase": "Super tower",
  "active": true,
  "members": [
    {
      "name": "Molecule Man",
      "age": 29,
      "secretIdentity": "Dan Jukes",
      "powers": [
        "Radiation resistance",
        "Turning tiny",
        "Radiation blast"
      ]
    },
    {
      "name": "Madame Uppercut",
      "age": 39,
      "secretIdentity": "Jane Wilson",
      "powers": [
        "Million tonne punch",
        "Damage resistance",
        "Superhuman reflexes"
      ]
    },
    {
      "name": "Eternal Flame",
      "age": 1000000,
      "secretIdentity": "Unknown",
      "powers": [
        "Immortality",
        "Heat Immunity",
        "Inferno",
        "Teleportation",
        "Interdimensional travel"
      ]
    }
  ]
}
```

FIG-24: JSON Sample

Week-6

(2.4.4) JavaScript

JavaScript is a language in which both interpreted and just-in-time compilation are possible. Because of its widespread usage on the Web, but also in a variety of other non-browser settings, such as Node.js, Apache CouchDB, and Adobe Acrobat, JavaScript is sometimes referred to as a scripting language.

Table-6: Data Types

Variable	Explanation	Example
String	This is a sequence of text known as a string. To signify that the value is a string, enclose it in single quote marks.	let myVariable = 'Bob';
Number	This is a number. Numbers don't have quotes around them.	let myVariable = 10;
Boolean	This is a True/False value. The words true and false are special keywords that don't need quote marks.	let myVariable = true;
Array	This is a structure that allows you to store multiple values in a single reference.	let myVariable = [1, 'Bob', 'Steve', 10]; Refer to each member of the array like this: myVariable[0], myVariable[1], etc.
Object	This can be anything. Everything in JavaScript is an object and can be stored in a variable. Keep this in mind as you learn.	let myVariable = document.querySelector('h1'); All of the above examples too.

the four main data structures that JavaScript supports are:

1. Arrays of sequential integers, such as ["one", "two", "three"]
2. { firstName: "Tom", lastName: "Smith" } are JS records, which are similar to JavaScript objects, such as { firstName: "Tom", lastName: "Smith" }
3. For instance, you may see something like the following when loading in a map: {"one":1, "two":2, "three":3}
4. To the entity tables, entity records are special maps that have an ID (or primary key) slot for each.

```

// program to format the date
// get current date
let currentDate = new Date();

// get the day from the date
let day = currentDate.getDate();

// get the month from the date
// + 1 because month starts from 0
let month = currentDate.getMonth() + 1;

// get the year from the date
let year = currentDate.getFullYear();

// if day is less than 10, add 0 to make consistent format
if (day < 10) {
    day = '0' + day;
}

// if month is less than 10, add 0
if (month < 10) {
    month = '0' + month;
}

// display in various formats
const formattedDate1 = month + '/' + day + '/' + year;
console.log(formattedDate1);

const formattedDate2 = month + '-' + day + '-' + year;

```

FIG-25: JavaScript Sample

Week-7

(2.4.5) CSS

In short, CSS is the programming that is used to style online content. The design of HTML is not a language of programming, as with CSS. You aren't using HTML either. Cascading Style Sheets (CSS) is a style sheet language. CSS is what you are using to style HTML elements based on their specific elements.



FIG-26: CSS

Table-7: Different Types of Selector

Selector name	What does it select	Example
Element selector (sometimes called a tag or type selector)	All HTML elements of the specified type.	p selects <p>
ID selector	The element on the page with the specified ID. On a given HTML page, each id value should be unique.	#my-id selects <p id="my-id"> or
Class selector	The element(s) on the page with the specified class. Multiple instances of the same class can appear on a page.	.my-class selects <p class="my-class"> and
Attribute selector	The element(s) on the page with the specified attribute.	img[src] selects but not
Pseudo-class selector	The specified element(s), but only when in the specified state. (For example, when a cursor hovers over a link.)	a:hover selects <a>, but only when the mouse pointer is hovering over the link.

Use of CSS is for web pages, and among other things, it incorporates design, layout, and various display preferences and standards for different screen sizes.

```
body {  
  background-color: lightblue;  
}  
  
h1 {  
  color: white;  
  text-align: center;  
}  
  
p {  
  font-family: verdana;  
  font-size: 20px;  
}
```

FIG-27:CSS Sample

(2.4.6) Bootstrap

Responsive, mobile-first web development is supported by the free and open-source Bootstrap framework. Design patterns including typography, forms, buttons, navigation, and other interface components may be found in this tool.

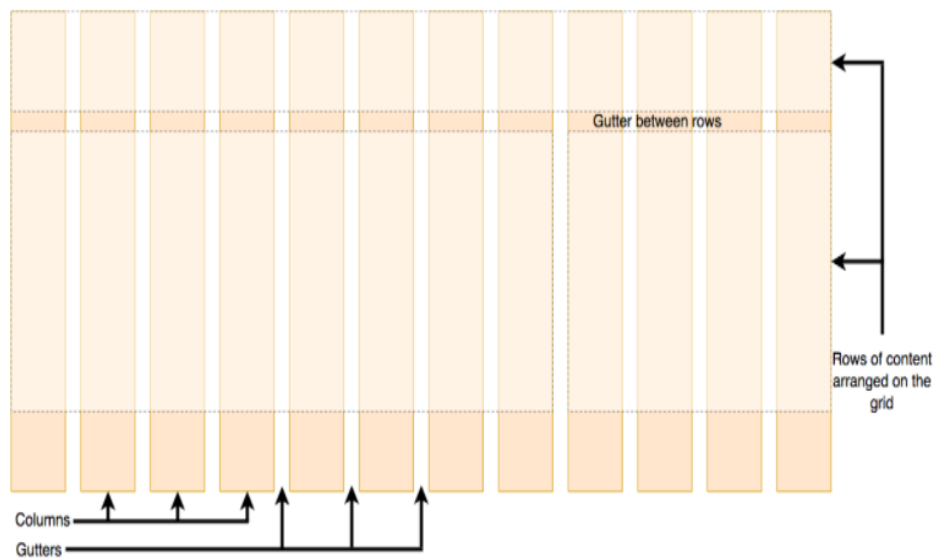


FIG-28: Grid System

```

<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<style>
/* Remove the navbar's default margin-bottom and rounded borders */
.navbar {
margin-bottom: 0;
border-radius: 0;
}
/* Set height of the grid so .sidenav can be 100% (adjust as needed) */
.row.content {height: 450px}
/* Set gray background color and 100% height */
.sidenav {
padding-top: 20px;
background-color: #f1f1f1;
height: 100%;
}
/* Set black background color, white text and some padding */
footer {
background-color: #555;
color: white;
padding: 15px;
}
</style>
</head>
<body>
<nav class="navbar navbar-inverse">
<div class="container-fluid">
<div class="navbar-header">
<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a class="navbar-brand" href="#">Logo</a>
</div>
<div class="collapse navbar-collapse" id="myNavbar">
<ul class="nav navbar-nav">
<li class="active"><a href="#">Home</a></li>
<li><a href="#">About</a></li>
<li><a href="#">Projects</a></li>
<li><a href="#">Contact</a></li>
</ul>
<ul class="nav navbar-nav navbar-right">
<li><a href="#"><span class="glyphicon glyphicon-log-in"></span> Login</a></li>
</ul>
</div>
</div>
</nav>
<div class="row content">
<!-- FILL NECESSARY CODE FOR COLUMN 1 HERE -->
<div class="col-sm-2 sidenav" id="col1">
<h4>Our Branches </h4>
<br>
<p>394 SW. Courtland Drive
Egg Harbor Township, NJ 08234
</p><br>
<p>76 East Middle River Street
Methuen, MA 01844</p>
</div>
<!-- FILL NECESSARY CODE FOR COLUMN 2 HERE -->
<div class="col-sm-8 text-left" id="col2">
<h4>Welcome</h4>
<p>You know, little of this, little of that. Do you have any kithua? I know how he likes to present himself; Father's weakness is vanity. Hence the plot. Please see him, Jeffrey. He's a good man. And thorough. They call Los Angeles the City of Angels. I didn't find it to be that exactly, but I'll allow as there are some nice folks there. "Course, I can't say I seen London, and I never been to France, and I ain't never seen no queen in her damn undies as the fella says. But I'll tell you what, after seeing Los Angeles and thisahere story I'm about to unfold wal, I guess I seen somethin' ever' bit as stupelyfn' as ya'd see in any a those other places, and in English too, so I can die with a smile on my face without feelin' like the good Lord cheated me. </p>
<hr>

</div>
<!-- FILL NECESSARY CODE FOR COLUMN 3 HERE -->
<div class="col-sm-2 sidenav" id="col3">
<h4>Categories</h4>
<div class="well">
<a href="#">Advice from Our Experts</a>
</div>
<div class="well">
<a href="#"> Our Blogs</a>
</div>
<div class="well">
<a href="#"> How to Use our Product</a>
</div>
</div>
Logo Home About Projects Contact Login

```

FIG-29: Bootstrap Sample

Week-8

(2.4.7) SQL

Microsoft SQL Server is a relational database management system created by Microsoft. The main role of a database server is to store and retrieve data for other software programmes as needed. Some of the SQL components include:

1. Statements and questions are made up of clauses, which are also constituents. This is (in some circumstances) optional.
2. In addition to expressions, which may return scalar numbers or tables made up of rows and columns of data, you may use expressions which yield more complex results.
3. Predicates, which define criteria that may be examined in SQL to determine whether the logic evaluates to 3VL (true/false/unknown) or Boolean truth values, and that are used to limit the impact of statements and queries, or to steer the course of execution.
4. A query is a tool used to search for data that you want, depending on certain parameters. In terms of SQL, this is a critical factor.
5. Persistent statements, which might regulate transactions, programme flow, connections, sessions, or diagnostics, or alter schemata and data.

(2.4.8) SQL Manipulation

The restrictions that are applied to the values of particular columns are called column constraints.

1. Using the PRIMARY KEY constraint allows you to identify a single row.
2. UNIQUE columns include values that are unique to each row.
3. Only columns with a value are required to be of type NOT NULL.
4. The DEFAULT column will have a default value if no value is supplied when it is created.

A new table is created in a database when you issue the command "CREATE TABLE". Incorporating this feature enables one to identify and label the table and each column inside it.

```
CREATE TABLE student (  
  id INTEGER PRIMARY KEY,  
  name TEXT UNIQUE,  
  grade INTEGER NOT NULL,  
  age INTEGER DEFAULT 10  
);
```

FIG-30: Create Table

Inserting new records (rows) into tables is done using the INSERT INTO command.

In other words, it has two distinct states:

1. Insert the content into the columns in the correct sequence.
2. Use "Name" columns instead.

```
-- Insert into columns in order:  
INSERT INTO table_name  
VALUES (value1, value2);  
  
-- Insert into columns by name:  
INSERT INTO table_name (column1, column2)  
VALUES (value1, value2);
```

FIG-31: Bootstrap Sample

To alter all columns of an existing table, use the ALTER TABLE statement. It is often used in combination with the Create COLUMN clause to introduce a new column.

```
ALTER TABLE table_name  
ADD column_name datatype;
```

FIG-32: Alter Table

To remove records (rows) in such a table, use the DELETE command. In this example, the WHERE clause is used to determine which record(s) should be destroyed. All records will be erased if the WHERE clause is removed.

```
DELETE FROM table_name  
WHERE some_column = some_value;
```


FIG-33: Delete Statement

(2.4.9) Queries

A SELECT query including the keyword SELECT* retrieves all columns from the table referenced in the result set. This query will return all of the movies' columns and records (rows).

```
SELECT *  
FROM movies;
```

FIG-34: Select Statement

Where is used to restrict the number of rows (records) that meet a certain criterion. Every records in which the pub year is 2017 will be returned by the query.

```
SELECT title  
FROM library  
WHERE pub_year = 2017;
```

FIG-35: Where Clause

You may use the BETWEEN operator to restrict your search to a certain range of values. Text, integers, or date data may be in the range of values. Any movie released between 1980 and 1990 will be found in the search results.

```
SELECT *  
FROM movies  
WHERE year BETWEEN 1980 AND 1990;
```

FIG-36: Between operator

You may use the ORDER BY clause to sort the result set in alphabetical or numerical order. Ordered in two ways: It may be ordered in two ways:

1. One may sort the results in decreasing order by using the keyword DESC.
2. ASC is a keyword used to sort the results in the order of increasing value (default)

```
SELECT *
FROM contacts
ORDER BY birth_date DESC;
```

FIG-37: Order By

You may use the AND operator to evaluate several criteria. A result set may only comprise records that meet both requirements connected by AND. This query will return results for any blue vehicle that was manufactured after 2014.

```
SELECT model
FROM cars
WHERE color = 'blue'
AND year > 2014;
```

FIG-38: AND Operator

Multiple criteria may be merged using the OR operator. Records containing either the AND or the OR in their search criteria are recorded in the returned result set.

```
SELECT name
FROM customers
WHERE state = 'CA'
OR state = 'NY';
```

FIG-39: OR Operator

(2.4.10) Aggregate Functions

Grouping entries in a sequence database by similar value over one or even more columns using the GROUP BY clause. When used with aggregate functions, it is often used to get data about entries that are comparable.

```
SELECT rating,
COUNT(*)
FROM movies
GROUP BY rating;
```

FIG-40: Group By Clause

The HAVING clause is being used to offer additional filter groups to the result set produced by the GROUP BY clause. When using aggregate functions in conjunction with aggregate functions, you may filter result sets depending on to an aggregate property.

```
SELECT year,  
       COUNT(*)  
FROM movies  
GROUP BY year  
HAVING COUNT(*) > 5;
```

FIG-41: Having Clause

Aggregate functions calculate a number on a group of values and return one number:

1. COUNT()
2. SUM()
3. MAX()
4. MIN()
5. AVG()

(2.4.11) Multiple Tables

Every record in a SQL database has a unique primary key field, since that field is used to identify each record. A main key must be populated..


 customer_id	f_name	l_name
1	Abby	Caren
2	Aaron	Paul
3	Gratian	Joseph

FIG-42: Primary

The foreign key is indeed a reference from one table's records to the table that contains. Foreign key functionality is crucial to retain numerous records for a given row.

customer_id	f_name	l_name
1	Abby	Caren
2	Aaron	Paul
3	Gratian	Joseph

order_id	customer_id	order_qty
1	2	5
2	2	6
3	1	2

FIG-43: Foreign key

UNION is used to collect data from numerous SELECT queries and eliminate duplicates.

```
SELECT name
FROM first_names
UNION
SELECT name
FROM last_names
```

FIG-44: Union

A CROSS-JOIN clause has been used to connect each row through one table to each row from another, and to do so, these clauses include specific rules that govern the contents of each result set row

```
SELECT shirts.shirt_color,
       pants.pants_color
FROM shirts
CROSS JOIN pants;
```

FIG-45: Cross Join Clause

Using an ON clause, results from even more than one tables may be combined to provide results based on their common column values.

```
SELECT *
FROM books
JOIN authors
ON books.author_id = authors.id;
```

FIG-46: ON Clause

Though the join requirement is not satisfied, rows from distinct tables are combined even if the join expression has been specified. The left join retrieves every row in the left table, but only if the join condition is fulfilled, which is why NULL values are used to fill in the fields from the right table

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

FIG-47: Join

The WITH clause stores the result of a query in a temporary table (temporary movies) using an alias. Multiple temporary tables can be defined with one instance of the WITH keyword.

```
WITH temporary_movies AS (
  SELECT *
  FROM movies
)
SELECT *
FROM temporary_movies
WHERE year BETWEEN 2000 AND 2020;
```

FIG-48: With Clause

CHAPTER-3: PERFORMANCE ANALYSIS

Performance of Code Challenges -

Java CC-1	75/100
Java CC-2	82/100
JavaScript CC	100/100
XML & JSON CC	100/100

Performance of Integrated Capability Test (ICT) –

Functional Testing	82/100
Java Programming Fundamentals	100/100

CHAPTER-4: CONCLUSION

My internship with Cognizant will finish on August 16, 2021, however over these eight weeks, I have increased my expertise in many technologies.

I'd want to thank all the coaches, SMEs, mentors, and trainers that assisted me throughout my internship at Cognizant. They helped me when I was having problems, and were there to handle all of my questions. My trainer, in particular, put in additional effort and served as an intern while working to pass on all of our inquiries to the corporation's higher authorities, including any queries about scheduling or workload.

My gratitude always go to TNP Officer Mr. Pankaj Kumar and Dr. Nafis U Khan, who went above and beyond to support and help with the whole placement effort.

REFERENCES

[1] Organization Handbook

[2] Organization Curriculum

[3] www.udemy.com

[4] www.geeks4geeks.com

[5] www.tutorialspoint.com

final project report

ORIGINALITY REPORT

17%

SIMILARITY INDEX

13%

INTERNET SOURCES

2%

PUBLICATIONS

12%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Jaypee University of Information Technology Student Paper	2%
2	Submitted to Sogang University Student Paper	2%
3	www.geeksforgeeks.org Internet Source	2%
4	www.codecademy.com Internet Source	2%
5	www.ir.juit.ac.in:8080 Internet Source	1%
6	www.nishorgo.org Internet Source	1%
7	tutorialspoint.dev Internet Source	1%
8	www.slideshare.net Internet Source	1%
9	www.scribd.com Internet Source	1%

10	en.wikipedia.org Internet Source	<1 %
11	Submitted to University of Wales Swansea Student Paper	<1 %
12	Submitted to Central Queensland University Student Paper	<1 %
13	Submitted to Gusto International College Student Paper	<1 %
14	Submitted to Modi Apollo International Institute Student Paper	<1 %
15	Submitted to CTI Education Group Student Paper	<1 %
16	Kishori Sharan. "Beginning Java 9 Fundamentals", Springer Science and Business Media LLC, 2017 Publication	<1 %
17	novatechsol.com Internet Source	<1 %
18	Submitted to Middlesex University Student Paper	<1 %
19	www.metaglossary.com Internet Source	<1 %
20	navinwbackup.blogspot.com Internet Source	<1 %

21	www.answers.com Internet Source	<1 %
22	www.onlinetutorial.co.in Internet Source	<1 %
23	data-flair.training Internet Source	<1 %
24	www.educba.com Internet Source	<1 %
25	"Beginning Java® Programming", Wiley, 2012 Publication	<1 %
26	codenza.app Internet Source	<1 %
27	github.com Internet Source	<1 %
28	text.123doc.org Internet Source	<1 %
29	tutorials.ducatinidia.com Internet Source	<1 %
30	www.coursehero.com Internet Source	<1 %
31	www.svecw.edu.in Internet Source	<1 %
32	www.vsbec.com Internet Source	<1 %

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 17-06-2021 ✓

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: Sarthak Pachauri Department: CSE Enrolment No 171308

Contact No:7807049958 E-mail: psarthak01@gmail.com

Name of the Supervisor: Mr. Surjeet Singh

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): INTERNSHIP REPORT

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

- Total No. of Pages =51
- Total No. of Preliminary pages =7
- Total No. of pages accommodate bibliography/references =1



(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at **17 (%)**. Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Abstract & Chapters Details	
	<ul style="list-style-type: none">• All Preliminary Pages• Bibliography/ Images/Quotes• 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Page counts	
			File Size	

Checked by

Name & Signature

Librarian

.....

Please send your complete Thesis/Report in (PDF) & DOC (Word File) through your Supervisor/Guide at plagcheck.juit@gmail.com

ORIGINALITY REPORT

17%

SIMILARITY INDEX

9%

INTERNET SOURCES

3%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Asian Institute of Technology Student Paper	4%
2	towardsdatascience.com Internet Source	3%
3	Submitted to Jaypee University of Information Technology Student Paper	5%
4	github.com Internet Source	1%
5	Satish Kumar Satti, K. Suganya Devi, Prasenjit Dhar, P. Srinivasan. "A machine learning approach for detecting and tracking road boundary lanes", ICT Express, 2020 Publication	1%
6	www.thushv.com Internet Source	1%
7	www.coursehero.com Internet Source	1%

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 17-06-2021 ✓

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: Sarthak Pachauri Department: CSE Enrolment No 171308

Contact No:7807049958 E-mail: psarthak01@gmail.com

Name of the Supervisor: Mr. Surjeet Singh

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): INTERNSHIP REPORT

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

- Total No. of Pages =51
- Total No. of Preliminary pages =7
- Total No. of pages accommodate bibliography/references =1



(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at **17 (%)**. Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.



(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Abstract & Chapters Details	
	<ul style="list-style-type: none">• All Preliminary Pages• Bibliography/ Images/Quotes• 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Page counts	
			File Size	

Checked by

Name & Signature

Librarian

Please send your complete Thesis/Report in (PDF) & DOC (Word File) through your Supervisor/Guide at plagcheck.juit@gmail.com