

STUDENT REPORT

On

Assessment of Student Behaviour in class tests through Data Visualization

Project report submitted in partial fulfilment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

By

UDISHA PANDEY

161469

UNDER THE GUIDANCE OF

**JAGDISH ,
INFOSYS PRIVATE LIMITED**



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

FEB-MARCH 2020

TABLE OF CONTENTS

DECLARATION...	<i>i</i>
ACKNOWLEDGEMENT.....	ii
LIST OF FIGURES.....	iv-v
LIST OF TABLES.....	vi
ABSTRACT... ..	vii
1. Chapter-1 INTRODUCTION	
1.1. Introduction... ..	2
1.2. Problem Statement.....	4
1.3. Objectives.....	5
1.4. Methodology.....	6
2. Chapter-2 LITERATURE SURVEY... ..	10-12
3. Chapter-3 SYSTEM DEVELOPMENT	
3.1. Model Development... ..	14-16
3.2. Design.....	47
3.3. Computational... ..	49
4. Chapter-4 PERFORMANCE ANALYSIS.....	50
5. Chapter-5 CONCLUSIONS	
5.1. Conclusions... ..	56
5.2. Future Scope.....	56
REFERENCES	59

DECLARATION

I hereby declare that the work reported in the B.Tech Project Report entitled **Student Report** submitted at **Jaypee University of Information Technology, Wagnaghat , India** is an authentic record of my work carried out under the supervision of **Mrs. Monika Bharti Jindal** . I have not submitted this work elsewhere for any other degree or diploma.



UDISHA PANDEY
161469

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.



Mrs. Monika Bharti Jindal
Associate Professor
Dept of Computer Science & Information Technology
Jaypee University Of Information Technology
Wagnaghat, Distt. Solan,
Himachal Pradesh

Date:22 June, 2020

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude to the Prof. Samir Dev Gupta (HoD), Department of Computer Science Engineering & Information Technology, Jaypee University of Information Technology, Solan for providing this opportunity to carry out the present work.

The constant guidance and consolation got from **Mrs. Monika Bharti Jindal** Associate Professor Department of Computer Science Engineering & Information Technology has been of incredible assistance in conveying our present work and helped us in finishing this task with success.

I would like to express a deep sense of gratitude to my trainers **Ms. Srijita** (SQL & C#) at Infosys Private Ltd. for their guidance and support in defining the design problem and towards the completion of my work. Without their wise counsel and able guidance, it would have been impossible to complete the thesis in this manner.

I am also thankful to our batch owners my batch owner **Mr. Jagdish**, for his intellectual support throughout the course of this work.

Finally, I thank my parents and my family members for giving the moral support and abundant blessings in all of activities and my dear friends who helped me to endure the difficult times with their unfailing support and warm wishes.

LIST OF FIGURES

S.No	Topic	Page
1	Data Visualization	9
2	Model Development	14
3	Data Hierarchy	15
4	Database System	16
5	Function of DBMS	16
6	Types of Database System	17
7	Hierarchical Database	18
8	Network Database	18
9	Relational Database	18
10	No Sql Database	19
11	Data Integrity	20
12	ER Model	20
13	Cross Foot Notation	21
14	Parts of SQL	22
15	Cross Join	23
16	Inner Join	24
17	Inner Join	25
18	Left Join	26
19	Right Join	27
20	Full Outer Join	28
21	Visual Studio	45
22	Form Design	48
23	App Interface	48
24	Student 1 test 1 score	50
25	Graph 1	51
26	Student 1 test 2 score	52
27	Graph 2	52
28	Student 1 test 3 score	53
29	Graph 3	53
30	Student 1 test 4 score	54
31	Graph 4	54

ABSTRACT

This project is an attempt to clarify how online multiple choice questions test can help to understand the behaviour and preparation of a student. During COVID-19 when it is going to be immensely difficult to prepare students to attend normal classroom programmes, it is going to be beneficial to develop an online method where students can give online tests in form of multiple choice questions and they can be evaluated by teachers automatically through an application which not only gives the score but also gives graphical output about how much time a student spends on each question and how it his/her behaviour while answering and marking different question. This monitoring is being done through data visualization. Whether the student answers a question or skips it and jumps to another and in what manner is he attempting the whole test will greatly reflect on the level of his understanding of the concept and level of his preparation. This paper helps to make students much better in any concept through data visualization of their behaviour of attempting the paper. This project reflects the importance of making an intuitive question paper so as to make students more attentive towards grasping a concept and revising it so that they become more efficient and score much better.

CHAPTER-1

INTRODUCTION

1.1 Introduction

In the world of technological advancement, the learning criteria and methods for students is modifying too. Earlier the only method to analyse the mental faculty of the student or to identify how much the student has understood was done through subjective questions and relatively big answers . But as times have changed due to shortage of time as well as vastness of the academic courses , multiple choice questions have reigned in the arena. Since they don't need human supervision, multiple choice questions have become extremely popular. But many a times multiple choice questions have been rendered as "closed-ended" . Many learners argue that it does allow them to show their full capacity. To solve this problem an experiment was carried out which was called "think out loud "method in which students had to narrate what they were thinking while answering the questions in the multiple choice questions . This allowed the assessors to get behind the thought process and mental ideas behind choosing among the many answers given . The teachers can use this data to analyse how much the student has actually learned through the methods of Knowledge discovery strategies. But this had to be non-invasive method. So in this project I have focussed on the solution which can enable the learning and recording of learner's habit while answering and selecting the right answers from the multiple choice questions . We can potentially analyse the data and help the learners become more reasonable in their preparation as well as becoming self dependent while giving the tests. In this world of fast technology, education need to be steady and effective . It is important for students to grasp the right concept and within the right framework of time . This project helps towards this motive using Data Visualization.

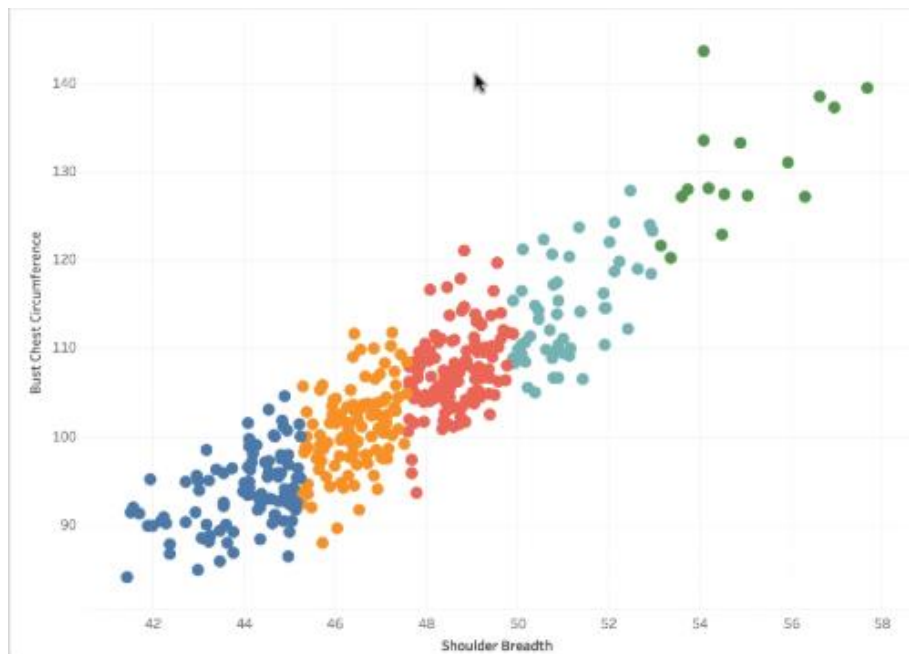
Data Visualization is the method in which we represent data in graphical form or in the form of chart or pie graphs etc so that we can extract useful patterns from it . When we have massive amount of data , we cannot simply work manually on the data . We need some device to find some meaningful patterns and useful knowledge which can lead to some meaningful research .

When we represent data using different colors and variety of patterns in lines , then it gets accustomed to find patterns and outliers and different trends.

In the education sector , the students and their learning process becomes the core which enables a wide variety of future for the youth . We can use different inputs of the students answering capabilities and understand their process of learning.

Applications of Data Visualization:

- Improved insights into complex data
- Faster decision making due to simplified graphic output
- Relate same information in different views.
- Form clusters of related and similar data .



Different methods of data visualization :

- Infographics
- Graphs
- Maps
- Dashboards
- Tables
- Charts

1.2 Problem Statement

A study was conducted to detect the significance of multiple choice question in schools. The outcome was flawed because it can't describe the actual student potential and contaminating important decisions. It also doesn't highlight the accuracy of the preparations of the student. Multiple choice questions are most commonly method to assess the score of students in the exams but it doesn't highlight how a student thinks and how much focus does a student give to each question. Why a student navigates between questions and why does the student go back to certain questions. These all parameters will be helpful in analysing how properly the student has grasped the concepts of the questions within the question paper. This paper tries to solve this problem and make multiple choice questions an effective way to analyse the thought process of students.

1.3 Objectives

The objective of this project is to determine the behaviour of a student while answering a particular multiple choice paper. There are many factors involved like taking a certain time to answer a particular question, switching between questions too fast or too slow, going back to revise at a very steady pace or haphazardly trying to answer the questions in no order just to pass the time inside the test. It also can analyse behaviour among different students attempting same questions at the same time, revising same questions, jumping between the two questions at the same time to analyse the pattern of cheating although it is included as a future scope of this project. This part of the project focuses solely on trying to analyse the behaviour of a student while attempting a particular set of questions in multiple choice type questions and try to understand the preparedness behind attempting certain questions and revising it many times.

1.4 Methodology

The proposed system uses the data of students using sql data base where we have stored the questions a student attempts, a question selected by the student and also the timestamp to collect the information about the time taken to attempt a questions. This sql data base will feed this data into an object oriented application which will analyse the data and try to produce a graph of performance of the student. This output will be visualized in the form of graph which will consist of questions attempted, question selected and the duration within which all this is being done. Here we have used

a SQL database connection which will have four attributes namely- questions selected, question attempted, timestamp , switched question.

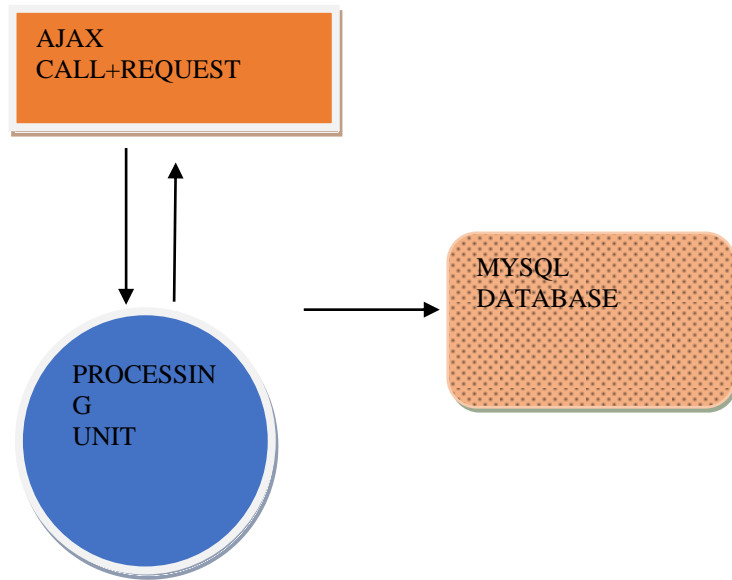


Figure : Model Development

CHAPTER-2

LITERATURE SURVEY

2.1 Behaviour During Examinations: A Comparison of 'A', 'C' and 'F' Students L McClain

In this paper L McClain has identified various strategies to find out what makes an A student different from F student . He focusses on various variables that determine whether a person is working hard enough for the test or not . The practices of revision , of jumping and witting between questions , changing the answers frequently all these variables comes into play while analysing a students attempts to the questions. The students were made to speak into recorder what they were thinking while attempting a particular question like what to anticipate in the question , if the urge to skip the question is more than attempting it , which questions the student wants to jump or if he wants to return back to this question later. So the psychology of the students was tested in this paper while attempting a multiple choice based paper .

2.2 Answer Changing Behaviour on Objective Examination John A. Bath

This paper examines the effect of answer changing on objective test performance . It compares the chances of the answer being correct when it is changed from one to another . This paper also took another angle of sexes wherein the females were more likely to change the answers than were males and the overall scores and performance of females was much better than males . This paper tried to identify the effect of changing the answers in the paper to the scores they get.

2.3 Information Visualization and Data Mining, D.A Mein

This paper reflects on the importance of visualization in exploring meaningful patterns between data . When there is a large dataset and we need to find some pattern or meaningful behaviour then we need some data mining techniques and visual techniques easily deal with homogenous and noisy data. It is intuitive and the stastical algors and complex mathematical parameters need not be understood in depth. This paper simplifies the aspects of Visual Data Mining.

2.4 Monitoring Online Tests through Data Visualization ,Gennaro Costalegio

This paper presents an approach on how to learn several important charactersticks of a student while he /she is giving a trest. The approach of logging the relevant data of a student and then examining the behaviorial pattern of the student while answering the questions is examined in this paper. Furthermore the data visualiation is focussed on in the form of charts , graphs and tables in order to give accurate description of the students .It focusses on Knowledge discovery, data visualization , distance learning and interactive data exploration.

CHAPTER-3

SYSTEM DEVELOPMENT

3.1 Model Development

Software Tools:

3.1.1 DBMS and SQL

- **WHAT IS DATA?**

Data is collection of facts about the object of interest. For e.g. data about an employee would include information like name, address, age, educational qualifications etc. Software Applications need to store data as it is required to answer a question e.g. how many employees are above 40 years of age? Data is also required to convey a story e.g. why are we successful as an organization?

Data is raw, only a lot of realities which without anyone else doesn't pass on anything. We have to comprehend designs between verifiable information and give it a significance. This is called data which causes us with answers to addresses like who, when, what, where and so forth. Combination of information and data drives us to respond to the how question and take business choices. This is alluded to as Knowledge.

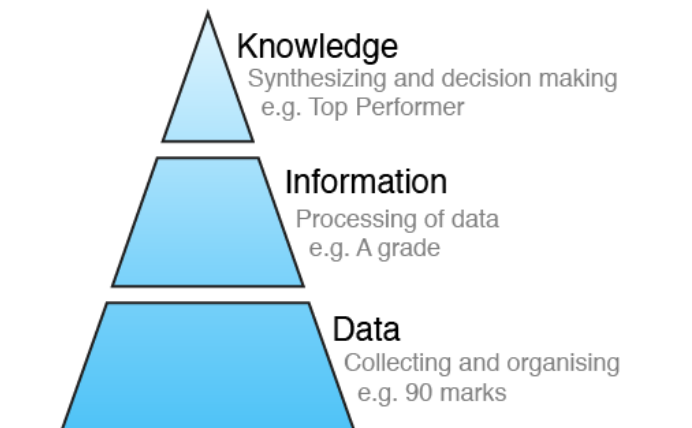


Figure 1.1 Data Hierarchy

- **REQUIREMENTS FROM DATA**

While using some software applications , data needs to have some properties be fulfilled in order to be useful.

Table 1.1 Requirements from Data

Requirement	Description
Integrity	Data should be accurate e.g. my facebook profile should contain valid <u>country name</u> .
Availability	I should be able to access facebook and see my data at all times.
Security	Only my friends should be able to see my posts and no one else.
Independent of Application	I should be able to access the same data from my Android app as well as <u>from web browser</u> on my laptop.
Concurrent	All my friends should be able to see my posts at the same time.

- **LIMITATIONS OF FLAT FILES**

Programming languages are used to access data which can be stored in flat files. The file based approach suffers following problems:

1. Data is stored in flat files and may be accessed using any programming language . The file based approach suffers following problems:
2. Dependency of program on body of knowledge
3. Complex process to retrieve data
4. Loss of knowledge on concurrent access
5. Inability to offer access supported record (Security)
6. Data redundancy

- **DATABASE SYSTEMS**

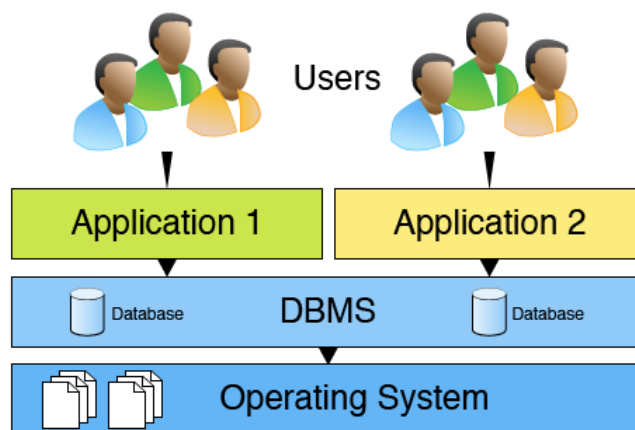


Figure 1.2 Database System

- **FUNCTIONS OF DBMS**

Several functions help us overcome associated file based systems in Database Management Systems .

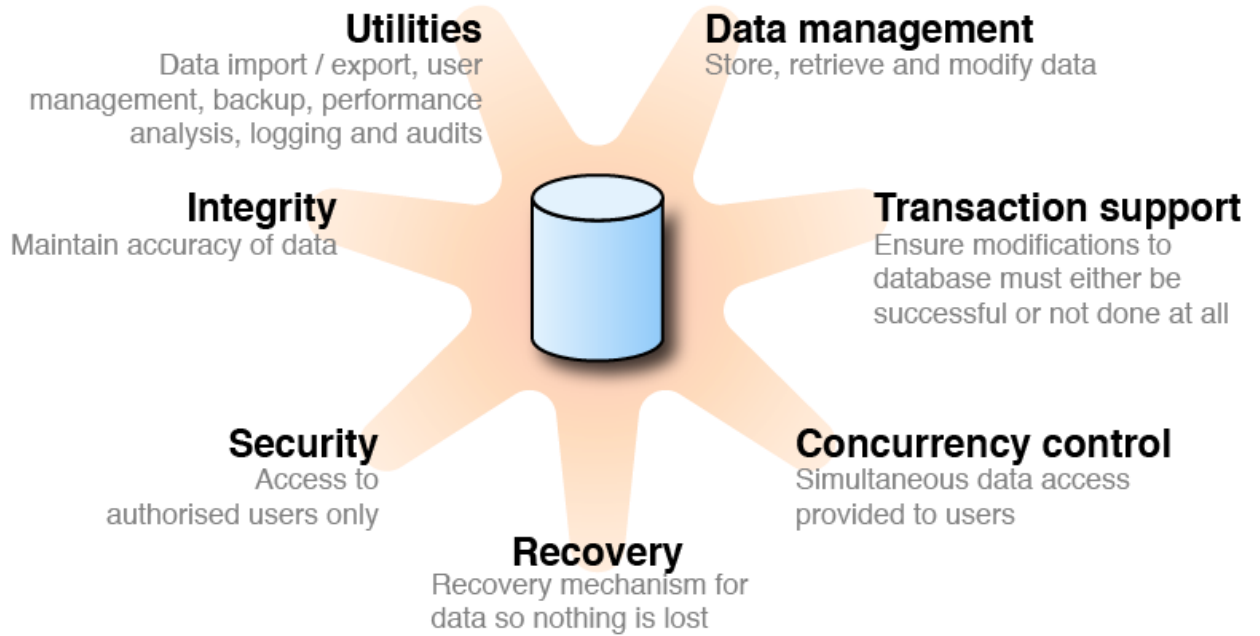


Figure 1.3 Functions of DBMS

• **TYPES OF DATABASE SYSTEMS**

We have four types of Database systems depending on what functionality we are trying to achieve. In the chronology of their emergence we have Hierarchical, Network, Relational and MySQL.

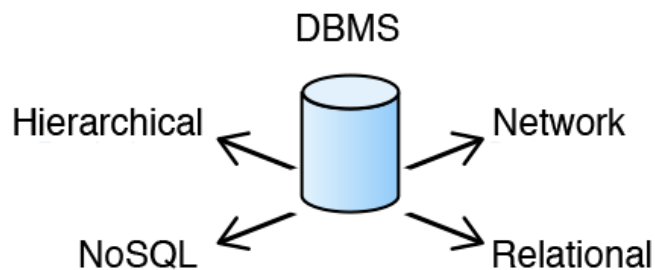
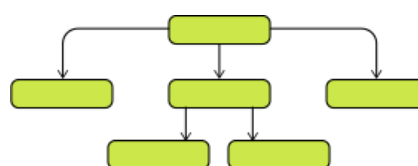
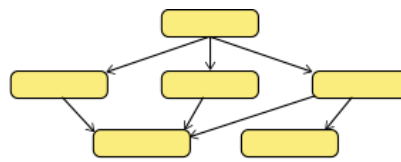


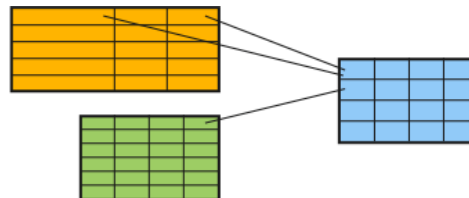
Figure 1.4 Types of Database Systems



Hierarchical
Tree structure



Network
Graph structure



Relational
Tabular structure

CANDIDATE KEY:- A Candidate Key can be any section or a mix of segments that can qualify as one of a kind key in database. There can be different Candidate Keys in a single table.

PRIMARY KEY:- To distinguish an interesting tuple inside a database table we need an primary key. The primary key ought to have these specific highlights:

Table 1.2 Mandatory and Desired attributes for a primary key

Mandatory	Desired
must uniquely identify a tuple	should not change with time
must not allow NULL values	<input type="checkbox"/> should have short size e.g. numeric data types

Sometimes one key is not sufficient to recognise a unique value and for that we need Composite primary key which is combination of two or more keys.

FOREIGN KEY:-A foreign key is a set of one or more columns in the child table whose values are required to match with corresponding columns in the parent table. Foreign key establishes a relationship between these two tables. Foreign key columns on child tables must be primary key or unique on the parent table. The child table can contain NULL values. Let us take Employee and Computer tables as provided below:

Parent / master / referenced table

Computer table

COMPID	MAKE	MODEL
1001	Dell	Vostro
1002	Dell	Precision
1003	Lenovo	Edge

Child / referencing table

Employee table

ID	ENAME	DEPT	COMPID
1	James Potter	ICP	1001
2	Ethan McCarty	ETA	NULL
3	Emily Rayner	ETA	1002

Figure 1.5 Foreign Key

Here we have Compid as the primary key in Computer table. Compid is foreign key in employee table which references from Computer table.

- **ENTITY RELATIONSHIP (ER) MODEL**

An Entity–relationship model (ER model) portrays the structure of a database with the assistance of an outline, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a structure or outline of a database that can later be executed as a database. The primary parts of E-R model are: element set and relationship set.

Table 1.3 ER Model Terms

Term	Definition	Examples
Entity	Real world objects which have an independent existence and about which we intend to collect data.	Employee, Computer
Attribute	A property that describes an entity.	Name, Salary

Here we have an example in which the entity is Employee and ID,ENAME,Salary,Dept are the attributes.

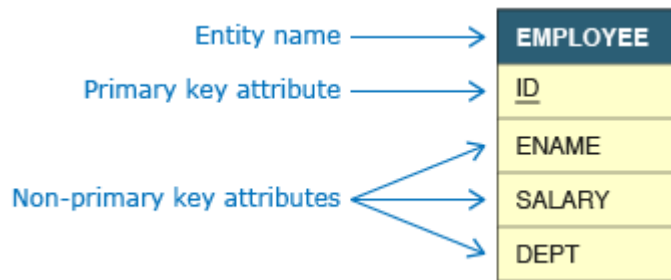


Figure 1.6 ER Model

To represent cardinality of relationship in an ER Model, we can use crow foot notation. We have four symbols and they represent different type of relations.

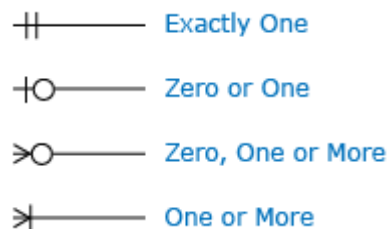


Figure 1.7 Cross Foot Notation

- **SQL**

Structured Query Language (SQL) is utilized to oversee information in every single social database like DB2, Oracle, SQL Server and so on. SQL principles are kept up by ISO. While most database items agree to the ISO standard, they likewise offer extra restrictive features.

Types of SQL Commands

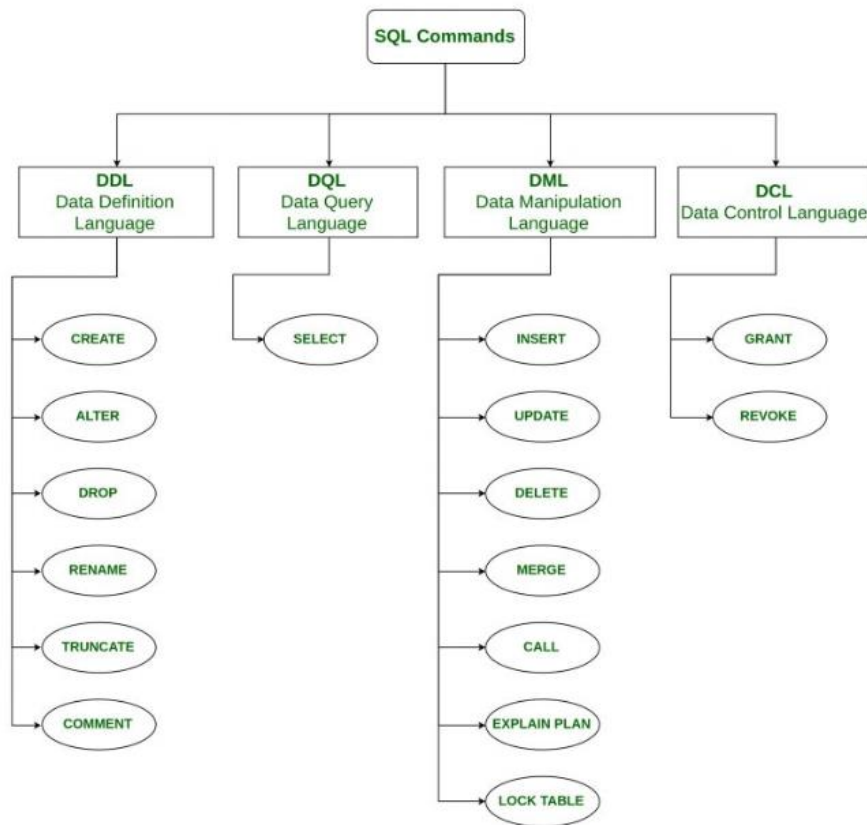


Figure 1.8 Types of SQL commands

1. **Data Definition Language** characterizes the pattern of a database which is social. It comprises of adjustment, creation, change and erasure of different items like put away techniques, imperatives, files. The yield is put away in metadata.
2. **Data Manipulation Language** is utilized to control or access a social database. It offers techniques to recover, make, delete, modify data in a database. A client is required to indicate what information is required.
3. **Data Control Language** permits clients to see put away strategies and tables in a social database. Whener a client is presented a question, the database check allowed privilege. If inquiry is unapproved then the entrance isn't conceded.
4. **Transaction Control Language** specifies commands for beginning and ending a transaction.

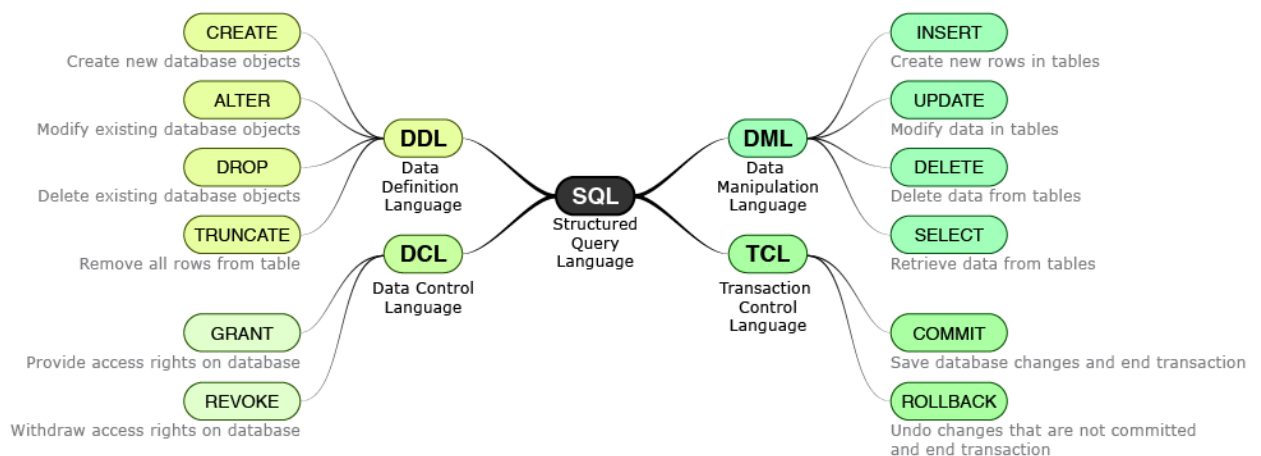


Figure 1.9 Parts of SQL

• DATA TYPES

There are three type of data types that SQL supports namely:

1. **SQL Character Data Types:** SQL supports two character data types for storing printable and displayable characters. They are used for storing information like name, address, description etc.**SQL Integral Data Types:** SQL supports SMALLINT, INTEGER and INT data types that are used for storing whole numbers. Unlike other databases Oracle does not define different size limits for them. They are all treated internally to have 38 digit of precision.
2. **SQL Non-Integral Data Types:** Nonintegral data types have an integer part and a fractional part. Either NUMERIC, DECIMAL or NUMBER data types can be used to store nonintegral numbers.

• DDL STATEMENTS

1. **CREATE TABLE** statement is used to make a table in a database. Database tables are composed into lines and segments. Each table will have a name which can have any number of segments (min 1 section is required). Every segment must have an information type which is the sort of qualities that can be put away. All tables must have a one of a kind name.

```
SYNTAX: CREATE TABLE Employee (  
EmployeeId INTEGER,  
EName VARCHAR2(10),  
Gender CHAR(1),  
DOJ DATE);
```

2. **DROP TABLE** statement is used to remove an existing table from the database.

```
SYNTAX: DROP TABLE Employee;
```

• CONSTRAINTS

Constraints are typically specified along with the CREATE TABLE statement. Constraints are arranged into different kinds dependent on the quantity of segments they follow up on just as in transit they are indicated.

Different constraints that can be made on database tables are:

1. **NOT NULL CONSTRAINT:** NOT NULL Constraint keeps a segment from tolerating NULL qualities. NOT NULL must be applied as a segment level requirement. Constraint name is discretionary and it very well may be determined by utilizing CONSTRAINT watchword.

```
SYNTAX: CREATE TABLE Employee(  
EmployeeId INTEGER CONSTRAINT Emp_EId_nn NOT NULL,  
EName VARCHAR2(15) NOT NULL,  
ELName VARCHAR2(15)
```

2. **DEFAULT CONSTRAINT:** A column can be given the default an incentive by utilizing DEFAULT alternative. The information sort of segment and default articulation must be the equivalent. DEFAULT choice can be accommodated nullable just as NOT NULL characteristics. Prophet database doesn't think about DEFAULT as a limitation.

```
SYNTAX: CREATE TABLE Employee (  
EmployeeId INTEGER,  
EName VARCHAR2(20),
```

DOJ DATE DEFAULT SYSDATE);

3. **PRIMARY KEY CONSTARINT**: PRIMARY KEY constraint on a column ensures that the column cannot contain NULL and duplicate values. We can have only one PRIMARY KEY in a table.

SYNTAX: CREATE TABLE Employee (
EmployeeId INTEGER CONSTRAINT emp_eid_pk PRIMARY KEY,
EFName VARCHAR2(15),
ContactNo NUMBER(10));

4. **CHECK CONSTRAINT**: CHECK constraint is used to limit the values that can be specified for a column.

SYNTAX: CREATE TABLE Employee (
EmployeeId INTEGER,
FName VARCHAR2(20),
Gender CHAR(1) CONSTRAINT Emp_gender_ck1 CHECK(Gender IN('M', 'F')));

5. **UNIQUE CONSTRAINT**: UNIQUE constraint on a section guarantees that two lines in a table cannot have same incentive in that segment. In contrast to Primary Key, UNIQUE imperative permits NULL qualities. A table can have numerous UNIQUE requirements.

SYNTAX: CREATE TABLE Employee (
EmployeeId INTEGER,
EFName VARCHAR2(20),
ContactNo NUMBER(10) CONSTRAINT Emp_cno_uk UNIQUE);

• OPERATORS

Arithmetic operators:

Operator	Symbol	Usage	Result
Addition	+	15 + 5	20
Subtraction	-	15 - 5	10
Multiplication	*	15 * 5	75
Division	/	15 / 5	3

Comparison Operator :

Operator	Symbol	Usage	Result
Equal to	=	15 = 5	false
Not equal to	<>	15 <> 5	true
Greater than	>	15 > 5	true
Greater than equal to	>=	15 >= 5	true
Less than	<	15 < 5	false
Less than equal to	<=	15 <= 5	false

Operator	Symbol	Usage	Example
Range	BETWEEN <lower limit> AND <upper limit>	Matches value between a range of values (Both inclusive)	Salary BETWEEN 2500 AND 3000
List	IN (List of values)	Matches any of a list of values	Dept IN ('IVS', 'ETA', 'ICP')
String pattern matching	LIKE	Matches a character pattern	SupplierId LIKE 'S%'
NULL Test	IS NULL	Is a null value	Bonus IS NULL

Logical Operator :

Operator	Symbol	Usage	Example
And	AND	Returns TRUE if both conditions are true	Salary >= 30000 AND Dept = 'ETA'
Or	OR	Returns TRUE if any one of the condition is true	Salary > 75000 OR Dept = 'ICP'
Not	NOT	Returns TRUE if following condition is false	Id NOT IN (2,3)

- **FUNCTIONS:**

Conversion Function:

Name	Syntax	Function
TO_CHAR	TO_CHAR(value, format)	Converts a number or a date to a string. Use this function for formatting dates and numbers.
TO_DATE	TO_DATE (value, format)	Converts a string to a date.
TO_NUMBER	TO_NUMBER (value, format)	Converts a string to a number.

Date Function:

Name	Syntax	Function
SYSDATE	SYSDATE	Returns current date of System i.e. the host on which database server is installed.
SYSTIMESTAMP	SYSTIMESTAMP	Returns current timestamp of the System.
ADD_MONTHS	ADD_MONTHS(date, n)	Adds n months to the given date.
MONTHS_BETWEEN	MONTHS_BETWEEN(date1,date2)	Finds difference between two dates in months.

CASE STATEMENT:

!.Searched Case Expression:

SQL CASE statement

CASE

```

WHEN Marks >= 85 THEN 'Excellent'
WHEN Marks >= 65 THEN 'Good'
ELSE 'Poor'

```

END

2.Simple Case Expression

SQL CASE statement

CASE Designation

```

WHEN 'SE' THEN salary * 1.2
WHEN 'SSE' THEN salary * 1.1
ELSE salary * 1.05

```

END

• ALTER TABLE

Alter is used to make modification inside a table with different types of command. Syntax for ALTER TABLE command is provided below:

	Clauses
Alter statement 1	ALTER TABLE Student ADD Address VARCHAR2(20)
Alter statement 2	ALTER TABLE Student MODIFY Address VARCHAR2(50)
Alter statement 3	ALTER TABLE Student RENAME COLUMN Address TO ResidentialAddress
Alter statement 4	ALTER TABLE Student DROP (ResidentialAddress)
Alter statement 5	ALTER TABLE Student ADD CONSTRAINT stud_sid_pk PRIMARY KEY (StudentId)
Alter statement 6	ALTER TABLE Student DROP CONSTRAINT stud_sid_pk

Figure 1. 9 ALTER STATEMENT

INSERTING DATA

INSERT STATEMENT: Insert statement is used to add tuples (records) to table. It supports three alternate syntax as shown below:

```

Table Name
↓
INSERT statement 1 INSERT INTO Employee
VALUES
(1, 'James Potter', '01-Jun-2014', 40000.00, NULL, 'FSI', 'TA', NULL, 1)

```

Figure 1. 10 INSERT STATEMENT

• **RETRIEVING DATA**

SELECT QUERY: SELECT query allows us to retrieve data from one or more tables in a relational database. It can be represented as:

	Attributes	Table name (Relation)	
Select query 1	SELECT *	FROM Employee	
Select query 2	SELECT Id, EName	FROM Employee	
Select query 3	SELECT Id, EName	FROM Employee	WHERE salary > 40000

↩ Required clauses ↪
Optional WHERE clause with conditions for selecting data

Figure 1. 11 SELECT STATEMENT

Use **DISTINCT** clause to remove duplicates. Usage of DISTINCT should be avoided as far as possible as it can lead to performance issues.

SYNTAX: SELECT DISTINCT Dept FROM Student

WHERE CLAUSE

The WHERE clause is used to filter records. The WHERE clause is used to extract only those records that fulfill a specified condition.

SYNTAX:

```

SELECT col1, col2, ...
FROM tables_name
WHERE condition;

```

LIKE OPERATOR

LIKE operator is used to match a character pattern. It allows us to use wild cards. SQL supports two wild cards: '%' which matches with any number of characters and '_' which matches with exactly one character.

SYNTAX:

```

SELECT col,col2, ...
FROM tables_name
WHERE columnN LIKE pattern;

```


- **UPDATING DATA**

UPDATE STATEMENT SYNTAX

Update statement is used to modify existing records in a single table in a relational database. Update statement can be represented as:

	Table name (Relation)		
UPDATE statement 1	UPDATE	Employee SET Salary = Salary * 1.1	
UPDATE statement 2	UPDATE	Employee SET Salary = Salary * 1.2	WHERE Id = 1
UPDATE statement 3	UPDATE	Employee SET Salary = Salary * 1.2, Bonus = 100	WHERE Id = 1

Required SET clause with attributes to be updated
Optional WHERE clause to filter rows

Figure 1. 12 UPDATE STATEMENT

The database system ensures that no constraints are violated during execution of an update statement. Any violation of constraints results in failure of the statement.

DELETING DATA

DELETE statement is utilized to erase records from a solitary table in a social database. The database framework guarantees that no limitations are abused during execution of an erase articulation. Any infringement of requirements brings about disappointment of the announcement.

TRUNCATE statement can likewise be utilized to erase information from tables. TRUNCATE statement erases all columns from the table as it doesn't bolster WHERE statement. Shorten articulation is a quicker choice contrasted with DELETE when we need to erase all columns from the

	Table name (Relation)		
DELETE statement 1	DELETE FROM	Employee	
DELETE statement 2	DELETE FROM	Employee	WHERE Dept = 'ETA'

Optional WHERE clause with conditions for selecting data

TRUNCATE statement
table. TRUNCATE TABLE Employee

Figure 1. 13 DELETE AND TRUNCATE STATEMENT

- **SQL FUNCTIONS**

SQL functions are built in modules provided by a database. We can use them in data manipulation statements to perform calculations on data. All functions return a single value.

NUMERIC FUNCTION

Numeric functions are single row functions that accept a numeric value and return numeric output.

Table 1. 4 Numeric Function

Name	Syntax	Function
ABS	ABS(value)	Returns absolute value of a number
ROUND	ROUND(value, digits)	Rounds the value to specified decimal digits
CEIL	CEIL(value)	Rounds up the fractional value to next integer

CHARACTER FUNCTION

Character functions work on character strings and can return a character string or a numeric value.

Table 1. 5 Character Function

Name	Syntax	Function
UPPER	UPPER(value)	Converts value to upper case
LOWER	LOWER(value)	Converts value in lower case
CONCAT	CONCAT(value1, value2)	Concatenates value1 and value2
LENGTH	LENGTH(value)	Returns the number of characters in value.

SUBSTRING FUNCTION

Substring function is used to extract part of a string. It has the following syntax SUBSTR(value, start_position, length)

<pre> 1 2 3 4 5 6 7 8 D A T A B A S E SUBSTR('DATABASE', 5) = 'BASE' </pre>	<pre> 1 2 3 4 5 6 7 8 D A T A B A S E SUBSTR('DATABASE', 3,3) = 'TAB' </pre>
---	--

Figure 1. 14 Substring Function

AGGREGATE FUNCTIONS

Aggregate functions work on numerous lines to restore a solitary line. Some total capacities like SUM (all out), AVG (normal) works just on numeric segments while others like MIN (most minimal worth), MAX (most noteworthy worth) and COUNT (number of lines) work on all information types. Every total capacity overlook NULL qualities aside from COUNT(*).

SYNTAX: SELECT MIN(Salary), MAX(Salary), SUM(Salary) FROM Employee

SYNTAX: SELECT COUNT(ID) COUNT_ID, COUNT(*) COUNT_STAR, COUNT(Bonus)
COUNT_BONUS FROM Employee

SYNTAX: SELECT AVG(Salary) AvgSalary, AVG(Bonus) AvgBonus1, SUM(Bonus) /
Count(Bonus) AvgBonus2 FROM Employee

• SORTING DATA

ORDER BY CLAUSE Order By clause is utilized to sort the aftereffect of a question in a specific request. Before we comprehend the punctuation of ORDER BY, Sorting on auxiliary section happens just when different lines have a similar incentive in the essential segment. The sort request can be distinctive for the two sections for example essential can be arranged in climbing and auxiliary in plummeting and the other way around. Request BY must be utilized to indicate the segments on which information must be arranged and the sort request for example climbing or dropping. Lines are arranged in climbing request if sort request isn't indicated. DESC ought to be utilized to sort the lines in descending order.

```
ORDER BY example 1  SELECT Id, EName, Salary FROM Employee ORDER BY EName
ORDER BY example 2  SELECT Id, EName, Salary FROM Employee ORDER BY EName, Salary
ORDER BY example 3  SELECT Id, EName, Salary FROM Employee ORDER BY EName DESC
ORDER BY example 4  SELECT Id, EName, Salary FROM Employee ORDER BY EName ASC, Salary DESC
ORDER BY example 5  SELECT Id, EName, Salary FROM Employee ORDER BY 2, 3
```

Figure 1. 15 ORDER BY CLAUSE

GROUPING DATA

GROUP BY

GROUP BY groups the data from the table into different groups based on criteria provided and calculates the aggregate function for each group. Thus the result has 1 row for each group.

SYNTAX: SELECT Dept, Designation, MAX(Salary) FROM Employee GROUP BY Dept, Designation;

HAVING

Having allows aggregate functions to be used as filter criteria which cannot be done using WHERE clause.

COMBINING DATA

UNION and UNION ALL clause to join results from at least two SELECT explanations. The select explanations might be from same or diverse tables. They must have same number of sections and their information types at same situation in both the inquiry must be good (either same or convertible through programmed change). Association expels all copies from the outcome. Two records are viewed as copies if values at comparing places of every one of their segments coordinate.

SYNTAX: SELECT CompId FROM Employee UNION SELECT CompId FROM Computer

• 1.8.5 JOIN CLAUSE

A JOIN clause is used to combine rows from two or more tables, based on a related column between them

INNER JOIN

The INNER JOIN keyword selects records that have matching values in both tables

Algorithm:

```
for each row r1 in Employee table
  for each row r2 in Computer table
    if r1.COMPID == r2.COMPID
      add combined row to Result
```

SYNTAX: SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;

Employee table:

ID	ENAME	DEPT	COMPID
1	James Potter	ICP	1001
2	Ethan McCarty	ETA	NULL
3	Emily Rayner	ETA	1002

Computer Table:

COMPID	MAKE	MODEL	MYEAR
1001	Dell	Vostro	2013
1002	Dell	Precision	2014

LEFT JOIN

The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.

Algorithm:

```
for each row r1 in Employee table
  set matched_in_comp to false
  for each row r2 in Computer table
    if r1.COMPID == r2.COMPID
      add combined row to Result
      set matched_in_comp to true
  if matched_in_comp is false
    add Employee row to Result
```

SYNTAX: SELECT column_name(s)
FROM table1

```
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

Employee table:

ID	ENAME	DEPT	COMPID
1	James Potter	ICP	1001
2	Ethan McCarty	ETA	NULL
3	Emily Rayner	ETA	NULL

Computer table:

COMPID	MAKE	MODEL	MYEAR
1001	Dell	Vostro	2013
1002	Dell	Precision	2014

Result

ID	ENAME	ECID	CCID	MODEL
1	James Potter	1001	1001	Vostro
2	Ethan McCarty	NULL	NULL	NULL
3	Emily Rayner	NULL	NULL	NULL

RIGHT JOIN

The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.

Algorithm:

```
for each row r1 in Employee table
  for each row r2 in Computer table
    if r1.COMPID == r2.COMPID
      add combined row to Result
      set matched_in_emp to true
for each row r3 in Computer table
  if not matched_in_emp is true
    add computer row to Result
```

SYNTAX:

```
SELECT column_name(s)
FROM table1
```

RIGHT JOIN table2
ON table1.column_name = table2.column_name;

Employee Table:

ID	ENAME	DEPT	COMPID
1	James Potter	ICP	1001
2	Ethan McCarty	ETA	NULL
3	Emily Rayner	ETA	NULL

Computer Table:

COMPID	MAKE	MODEL	MYEAR
1001	Dell	Vostro	2013
1002	Dell	Precision	2014

Result:

ID	ENAME	ECID	CCID	MODEL
1	James Potter	1001	1001	Vostro
NULL	NULL	NULL	1002	Precision

• 1.8.6 SUBQUERY

Subquery is an inquiry inside a question. A subquery must be encased in sections and can be utilized in SELECT, FROM, WHERE and HAVING conditions. Subquery in SELECT and FROM proviso are once in a while utilized. Subqueries in WHERE and HAVING provisos are arranged into Independent and Correlated subqueries.

INDEPENDENT SUBQUERY

In an independentsubquery, the internal and external inquiry are autonomous of one another. We can run an internal question and assess its outcome autonomous of the external inquiry. independent subquery are additionally arranged into single column and various line types relying on the quantity of lines returned.

```
SELECT ID, EName, salary FROM Employee A WHERE salary
= (SELECT MAX(salary) FROM Employee B)
```

Figure 1. 16 Independent Subquery

CORRELATED SUBQUERY

A Correlated subquery is one in which the internal inquiry that relies on the external question for it's execution. Explicitly it utilizes a segment from one of the tables in the external question. The internal question is executed iteratively for each chose column of the external inquiry. If there should arise an occurrence of free subquery, the inward question just executes once.

```

SELECT Id, Ename, Designation, Salary FROM Employee E1
WHERE Salary >= (SELECT Avg(Salary) FROM Employee E2 WHERE E1.Designation = E2. Designation);

```

The column of table present in outer query (Employee E1) is used inside the inner query (E1.Designation)

Figure 1. 17 Correlated Subquery

3.2 C# .NET

C# is object oriented language which is spelled as C-Sharp and it is a general purpose language. Anders Heljsberg along with his Microsoft team developed this language within the DOTNET initiative. It is a lot identical to JAVA and has object oriented programming principles.

.NET framework is a Windows-only version of .NET which is used for building any type of application that can be run on Windows. It began as a proprietary software although that software stack was standardized immediately .It targets alternating operating system, plug-ins, embedded devices, mobile computing.

The first framework of .NET was released in 2002. Almost 60 programming languages are supported by .NET FRAMEWORK. It is platform dependent and can run only on Windows based operating system. To use the .NET framework , we need visual studio which is a development tool which is used to design the .NET framework .

Framework Class Library:

The framework class library has many uses:

1. Wrapper Around OD usefulness: They incorporate utilization of record framework, arrange highlights , classes to deal with I/O applications.
2. Frameworks: There are different structures accessible in the FCL to build up specific applications. For instance, ASP.NET is utilized to create web applications, Windows Presentation Foundation (WPF) is utilized to render UIs in Windows applications, etc.

3. **Utility Features:** The utility features in the FCL incorporates different assortment classes, for example, list, stack, line, word reference, and so on and furthermore classes for progressively shifted controls, for example, Regex class for normal articulations.

Some frameworks are given below:

System	The System namespace has base classes for definition of interfaces, data types, events, event handlers, attributes, processing exceptions etc.
System.Activities	The System.Activities namespace handles the creation and working with activities in the Window Workflow Foundation using various classes.
System.Collections	The System.Collections namespace has multiple standard, specialized, and generic collection objects that are defined using various types.
System.Configuration	The System.Configuration namespace handles configuration data using various types. This may include data in machine or application configuration files.
System.Data	The System.Data namespace accesses and manages data from various sources using different classes.

• **FEATURES OF C #:**

1. **Easy to start:** C# is a significant level language so it is nearer to other well known programming dialects like C, C++, and Java and in this way turns out to be anything but difficult to learn for anybody.
2. **Widely used for developing Desktop and Web Application:** C# is generally utilized for creating web applications and Desktop applications. It is one of the most well known dialects that is utilized in proficient work area. On the off chance that anybody needs to make Microsoft applications, C# is their first decision.
3. **Community:** The bigger the network the better it is as new devices and programming will be creating to improve it. C# has an enormous network so the advancements are done to make it exist in the framework and not become terminated.
4. **Game Development:** C# is broadly utilized in game turn of events and will keep on ruling. C# incorporates with Microsoft and in this way has a huge objective crowd. The C# highlights, for example, Automatic Garbage Collection, interfaces, object-situated, and so forth make C# a well known game creating language.

• **ADVANTAGES OF C#:**

1. C# is effective in dealing with the framework. All the trash is consequently gathered in C#.
2. There is no issue of memory spill in C# in light of its high memory reinforcement.

3. Cost of support is less and is more secure to run when contrasted with different dialects.
4. C# code is ordered to a moderate language (Common (.Net) Intermediate Language) which is a standard language, freely independent of the objective working framework and design.

• **C SHARP VARIABLE:**

It is the name given to memory location. It can be modified and can be used again whenever needed.

Table Variables

Variable Type	Example
Decimal types	decimal
Boolean types	True or false value, as assigned
Integral types	int, char, byte, short, long
Floating point types	float and double
Nullable types	Nullable data types

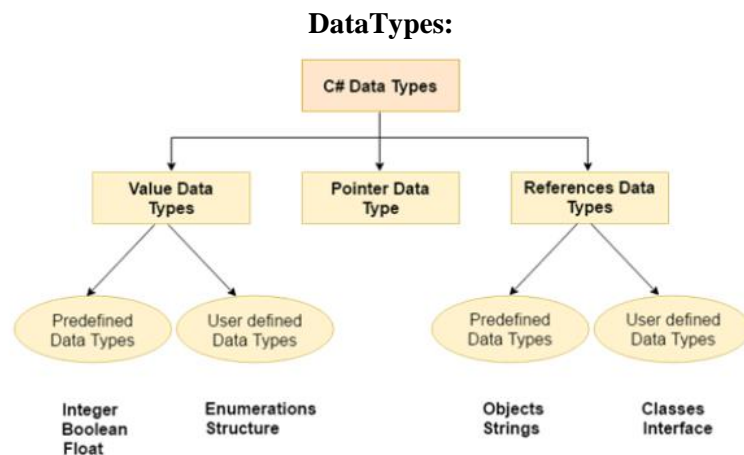


Figure Data Types

Table Data Types

Data Type	Size	Range
char	1 byte	-128 to 127
signed char	1 byte	-128 to 127
unsigned char	1 byte	0 to 127
short	2 byte	-32,768 to 32,767
signed short	2 byte	-32,768 to 32,767

unsigned short	2 byte	0 to 65,535
int	4 byte	-2,147,483,648 to 2,147,483,647
signed int	4 byte	-2,147,483,648 to 2,147,483,647
unsigned int	4 byte	0 to 4,294,967,295
long	8 byte	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
signed long	8 byte	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

• **OPERATORS:**

It is used to perform operations. We have arithmetic , logical , relational operators etc.

They are mentioned in table below:

Table Operators

Category	Operator	Associativity
Unary	~ ++ -- (type)* & sizeof	Right to Left
Multiplicative		Right to Left
Additive		Right to Left
Relational	< > = >=	Right to Left
Shift	>	Right to Left
Equality	=	Right to Left
Bitwise AND		Right to Left
Bitwise OR		Right to Left
Bitwise XOR		Right to Left
Bitwise NOT		Right to Left
Bitwise AND		Right to Left

• **DECISION MAKING STATEMENTS:**

It is similar to when we take life decision with different conditions and different consequences. A program can control flow of execution by using control statements.

If statement:

```
if(condition)
{
//code to be executed }

```

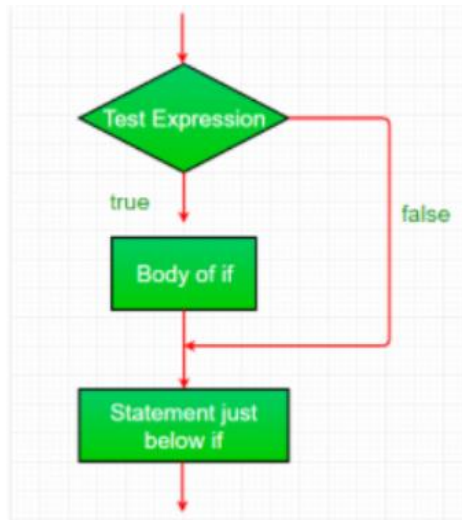


figure if statement

If-else statement:

```
if(condition)
{ //code for true condition
}
Else
{
//code if condition is false
}

```

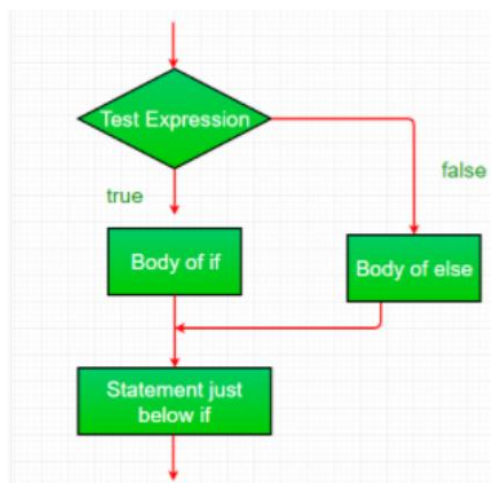


Figure If Statement

Switch Statement:

It gives a range of values to select from various conditions.

Syntax:

```
switch (expression)
{
case value1: // statement sequence
    break;
case value2: // statement sequence
    break;
.
.
.
case valueN: // statement sequence
    break;
default: // default statement sequence
}
```

Foreach Loop :

It is used to iterate over elements present inside a collection.

It may be a array or a list .

Syntax:

```
foreach(data_type var_name in collection_variable)
{
    // statements to be executed
}
```

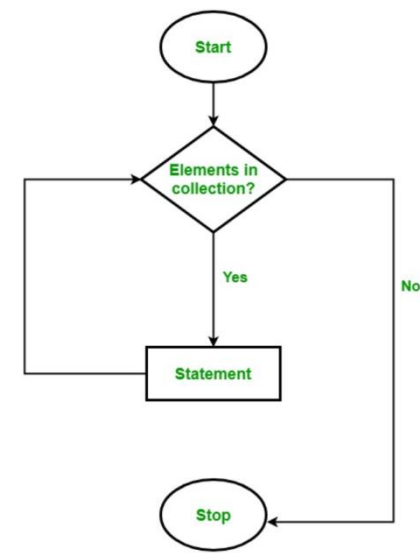


Figure foreach loop

- **CLASSES AND OBJECTS:**

Class and Object are the essential ideas of Object-Oriented Programming which spin around the genuine substances. A class is a client characterized outline or model from which articles are made. Fundamentally, a class consolidates the fields and methods(member work which characterizes activities)

into a solitary unit. In C#, classes bolster polymorphism, legacy and furthermore give the idea of inferred classes and base classes.

For the most part, a class presentation contains just catchphrase class, trailed by an identifier(name) of the class. Yet, there are some discretionary traits that can be utilized with class affirmation as indicated by the application prerequisite. When all is said in done, class announcements can incorporate these parts, all together:

1. Modifiers: A class can be open or inner and so on. Of course modifier of class is inner.
2. Keyword class: A class catchphrase is utilized to pronounce the sort class.
3. Class Identifier: The variable of type class is given. The identifier(or name of class) should start with an underlying letter which ought to be promoted by show.
4. Base class or Super class: The name of the class' parent (superclass), assuming any, went before by the : (colon). This is discretionary.
5. Interfaces: A comma-isolated rundown of interfaces actualized by the class, assuming any, went before by the : (colon). A class can execute more than one interface. This is discretionary.
6. Body: The class body is encircled by { } (wavy supports).

- **OBJECTS**

It is a fundamental unit of Object-Oriented Programming and speaks to the genuine substances. A normal C# program makes numerous articles, which as you probably are aware, collaborate by summoning strategies. An item comprises of :

1. State: It is spoken to by characteristics of an article. It likewise mirrors the properties of an item.
2. Behavior: It is spoken to by strategies for an article. It likewise mirrors the reaction of an item with different articles.
3. Identity: It gives a novel name to an item and empowers one article to communicate with different articles.

- **CONSTRUCTOR OVERLOADING:**

It is the capacity to reclassify a Constructor in more than one structure. A client can actualize constructor over-burdening by characterizing at least two constructors in a class having a similar name. C# can recognize the constructors with various marks. for example the constructor must have a similar name yet with various boundaries list.

We can over-burden constructors in various manners as follows:

1. By utilizing distinctive sort of contentions
2. By utilizing diverse number of contentions
3. By utilizing distinctive request of contentions

By utilizing distinctive sort of contentions

Example:

```
public ADD (int a, float b);  
public ADD (string a, int b);
```

By utilizing diverse number of contentions

For this situation, we will utilize at least two constructors having the distinctive number of boundaries. The information sorts of contentions can be the equivalent however the quantity of boundaries will be unique.

Example:

```
public ADD (int a, int b);  
public ADD (int a, int b, int c);
```

By utilizing distinctive request of contentions

Example:

```
public student(double a, int x, string s)  
public student(string s, int x, double a)
```

INVOKE AN OVERLOADED CONSTRUCTOR USING “THIS” KEYWORD

We can call an over-burden constructor from another constructor utilizing this watchword however the constructor must be have a place with a similar class, since this catchphrase is pointing the individuals from same class wherein this is utilized. This kind of calling the over-burden constructor likewise named as Constructor Chaining.

Example:

Let the class name is c1,

Now

```
public c1()  
public c1(int a) : this()  
public c1(double b) : this(int)
```

• METHODS:

They are block of programs designed to give the ability to user to reuse some special functions. It is a time saver and improves the readability of the code. A method can or cannot return anything depending on the return type.

Syntax:

<Access modifier> <return type> <method name> ([<param_list>])

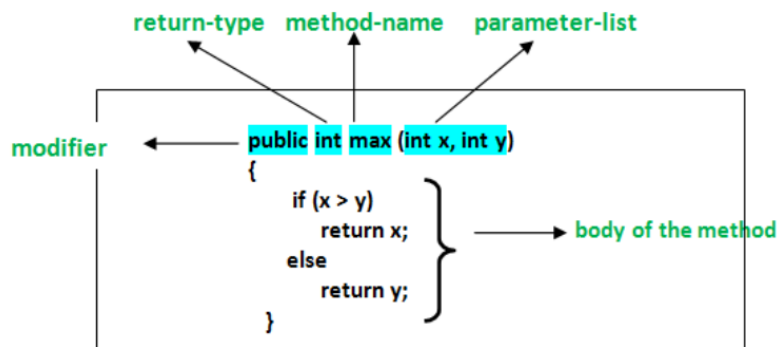


Figure methods

- **REF AND OUT PARAMETERS:**

Ref keyword is used to pass arguments as a reference type. It passes reference rather than a copy of the value which is original.

Out keyword used pass method arguments as reference type.

The main difference between both out and ref is that in ref parameter is necessary to be initialized while its not necessary in out.It is also not mandatory to initialize value of the parameter before returning to the calling method and opposite in out.

- **BOXING AND UNBOXING:**

When we convert value type to reference type object then it is known as boxing. Boxing is implicit conversion.

Eg: int number=23;

Object obj=num;

Here we have a value type number which stores 23 , while assigning it to a object obj , we copy and store it in the heap .

When we convert reference type object into value type then it is known as Unboxing and it is and explicit conversion.

Example:

Int num=23;

Object obj=num;

Int i=(int)Obj;

Here we have to do an explicit conversion of an object type variable into integer type.

PARAMS:

It is useful because it can act aas a parameter that can take many number of arguments in it.

When we dont know about the number of parameters to be used in the code we can just use params.

After the declaration of params keyword , no additional params can be allowed.

If the length of arguements passed is 0 then the params is empty.

Example:

```
Class Param_concept
{
Static void Add(params int[] List_of_arguements)
{
//body of function
}
}
```

- **PROPERTIES IN C#:**

1. We use properties for basic two reason:

- If we dont have public members and only private members , then properties enable the program to read, write or compute those private fields.

- if the members are public then there is possibility of misuse.

2. We make use of get and set accessors to do this task.

3. The getter is used to obtain the value of the private field from the user or console .

4. The set method is used to set that same value to the specific method.

5. When we have bot get and set method then it is read and write properties.

6. When we have only get method then it has read only properties.
7. When we have only set method then we have write only properties.
8. When there is no insertion of get and set method then it is auto implemented properties.

Syntax:

```
<access modifier> <return type> <property name>
{
  Get
  { // body
  }
  Set
  {
  //body
  }
```

- **STATIC KEYWORD:**

Static keyword is applicable in 4 ways :

STATIC CLASS:

A static class can have only static methods, data members and constructor.

A static class cannot have object.

They cannot inherit a static class from another class so they are also called sealed.

STATIC VARIABLE:

A static variable has only a single copy which is shared among all objects of a class. Any change made in a static variable is reflected all over the objects.

A static object can be accessed with the name of the class, it doesn't need any object to be accessed.

STATIC METHOD:

A static method is accessed directly through the class, they don't need any object.

It can have both non static and static fields.

STATIC CONSTRUCTOR :

They are invoked only once in the class and it is invoked to initialize the static member of the class. Of all the constructors, the static constructor is invoked before the first constructor is run. It is executed only once. It is not in user's control to invoke the static constructor.

Indexer: It is a feature of C sharp that allows struct or an instance of a class to be indexed as an array. The class which is defined as an indexer then it will behave like a virtual array. We use [] operator to access the instances of class. The accessors of indexer take parameters as opposed to Properties.

Syntax:

```
[access_modifier] [return_type] this [argument_list]
{
  get
```



```

{
    // get block code
}
set
{
    // set block code
}
}

```

Example:

Class Indexer_class

```

{
Public int this[int index]
{
Get
{
Return val[index]; ;; retrieving values stored in val[] array.
}
}
Set
{
val[index]=value;
}
}

```

The indexers can be overloaded.

They are not same as properties.

This keyword is used to declare an indexer.

Since it is an instance member of class that is why cant be a static member.

• INHERITANCE

It is similar concept to that of Object Oriented Programming. By this mechanism one class can inherit the properties from another class.

Super Class: The class from which the properties are inherited.

Derived Class: The class which inherits the properties from the super class is derived class.

Reusability: We can reuse the features inside a base class by inheriting from it in another class.

Syntax:

```

Class super
{
//body of class
}
Class derived : super
{
//body of derived class
}

```

We have several types of inheritance in C sharp.

1.Single Inheritance: One class inherits another class

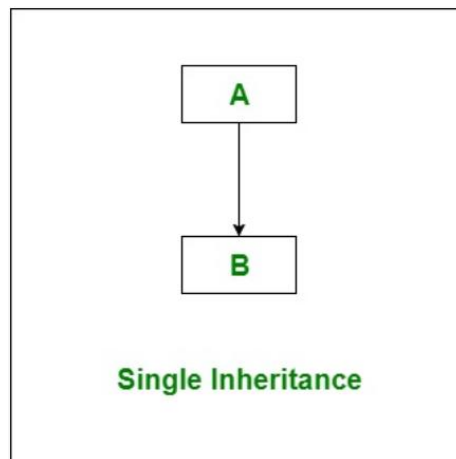


Figure single inheritance

2. Multilevel inheritance:

A derived class inherits from another class which is in turn inherited by some super class.

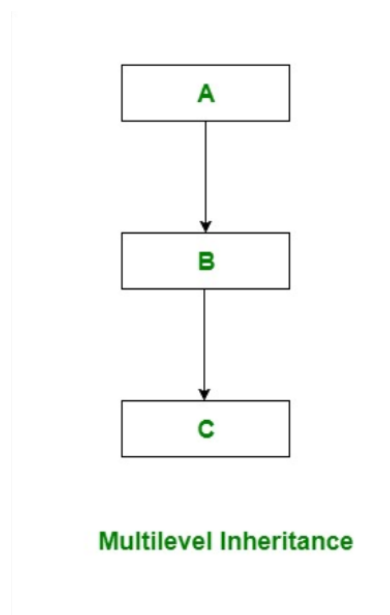


Figure multilevel inheritance

3.Heirarchichal Inheritance: When more than one class is inherited from a super class.

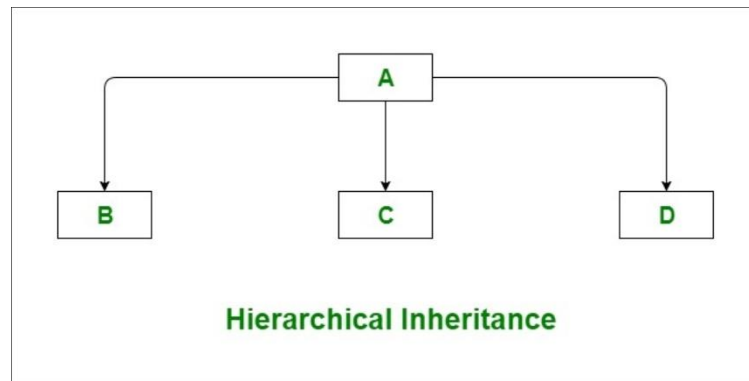


Figure heirarchial inhertance

4. Multiple Inheritance: When one class inherited by more than one super class. But this type of inheritance is not possible in C sharp but it can be implemented using interface.

INHERITANCE IN CONSTRUCTORS:

Constructors can be present in both the super class as well as derived class. The derived class dosent inherit the constructor of the base class . It is allowed only to invoke the constructor of the base class.

We can have two cases:

Case 1: When only derived class contains a constructor , in this case the object sof base class are instantiated automatically by the default constructor.

Case 2: When base and derived class both have a constructor , then with the help of base keyword , the derived class can instantiate the objects of base class in order to instantiate it's own objects.

Syntax:

```

derived-constructor(parameter-list) : base(argument-list)
{
    // body of constructor
}
  
```

- **METHOD OVERRIDING:**

It is a method to allow invoking of functions from another base class in the derived class. When the derived class method has same name as base class method then it is called method overriding.

It is feature that allows derived class to have it's own implementation of a method. This is run time polymorphism.

Syntax:

```

Class base_class
{
    Void function_base_class();
}
Class derived_class:base_class
{
  
```

```

Void function_base_class();
}
Class main_method
{
main()
{
Derived_class obj=new derived_class();
obj.function_base_class();
}
}

```

Here the base class is inherited by the derived class and the derived class is implementing the method of the base class by overriding the function of the base class.

Virtual keyword: It is used within base class . It is used to change in base class for overridden that particular method in derived class.

Override: It is used inside derived class method. It modifies virtual method into derived class .

Base: this keyword is used to access member of base class from derived class.

It cannot be used within a static method.

• EXCEPTION HANDLING:

Exception Handling is used to handle unwanted interruptions in a code. The normal flow of the program continues with the help of exceptions.

It is an event which is thrown at runtime. They are derived from System.Exception class. We have various inbuilt exception class in C sharp like :

Exception	Description
System.DivideByZeroException	handles the error generated by dividing a number with zero.
System.NullReferenceException	handles the error generated by referencing the null object.
System.InvalidCastException	handles the error generated by invalid typecasting.
System.IO.IOException	handles the Input Output errors.
System.FieldAccessException	handles the error generated by invalid private or protected field access.

We have four keyword in handling exceptions:

- 1.try
- 2.catch
- 3.finally
- 4.throw

Try: it is used to contain block which may have an exception.

Catch: it handled the exception that is thrown by try block.

Syntax:

```

Try
{
//body of try
}

```

```

Catch
{
//body of catch
}

```

Finally:

It is used to execute important part of code whether the exception thrown by a block of code is handled or not. It should be preceded by a try -catch block.

Syntax:

```

Try
{
//body
}

```

```

Catch
{
//body
}

```

```

Finally
{
//body
}

```

USER DEFINED EXCEPTIONS

The user can define their own exceptions, for which we need to inherit Exception class.

The throw keyword is used to throw a user defined exception.

Syntax:

```

Class User_defined_exception: Exception

```

```

{
Public some_exception(SString message):base(message)
    {
    }
}

```

```

Public class test_user_defined_exception

```

```

    {
    Throw new some_exception("We have thrown an user defined exception");
}

```

```

main()

```

```

{
Try
{
//body of try
}
catch(some_exception e )
{
//body
}
}

```

```
}
```

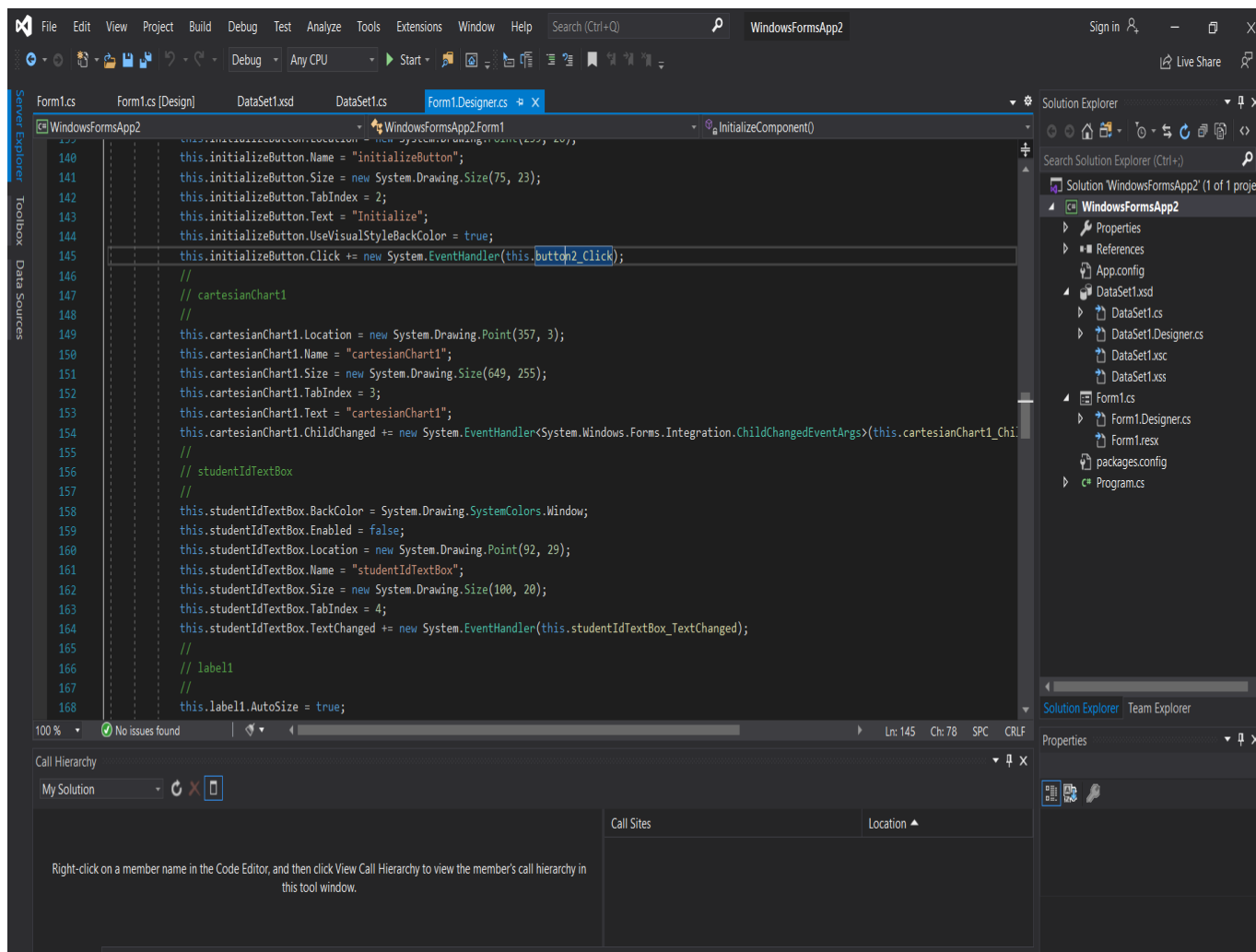
3.3 VISUAL STUDIO

It is a software from Microsoft which is used for integrated environment for development. It can host web services , make web apps , websites, computer programs. It uses various platforms like Microsoft Silverlight, Windows Store , Windows Forma.

It supports a code editor which is called intellisense which can help in code completion component . It also has code refactoring service . It works on both level- source level and debugger level. It also consist of a code profiler web designer , data schems designer.

It support around 36 different types of languages and here in this project C sharp has been used to implement the object oriented details of input .

Varioud languages like Java Script , Type Script , C # , Java , Visual Basic , HTML, CSS etc can also be implemented.



3.4 Design:

To start with the design of the project , I made a class of Student Activity which would consist of the various things an individual can do in a test . It included of :

- StudentId
- TestId
- TimeStamp
- QuestionSelected
- AnswerSelected
- SelectionValue
- QuestionNumber

The StudentId will be unique for each individual .The TestId will be unique for each student .Here we have assumed that the student can give many tests.

The timestamp calculates to duration of whole tests or to know which question is attempted at what time .

The QuestionSelected will determine after initialization which question is selected by the student and which answer is selected is stored in AnswerSelected.

The selection Value stores among “a,b,c,d” depending on what value the student thinks is the correct answer.

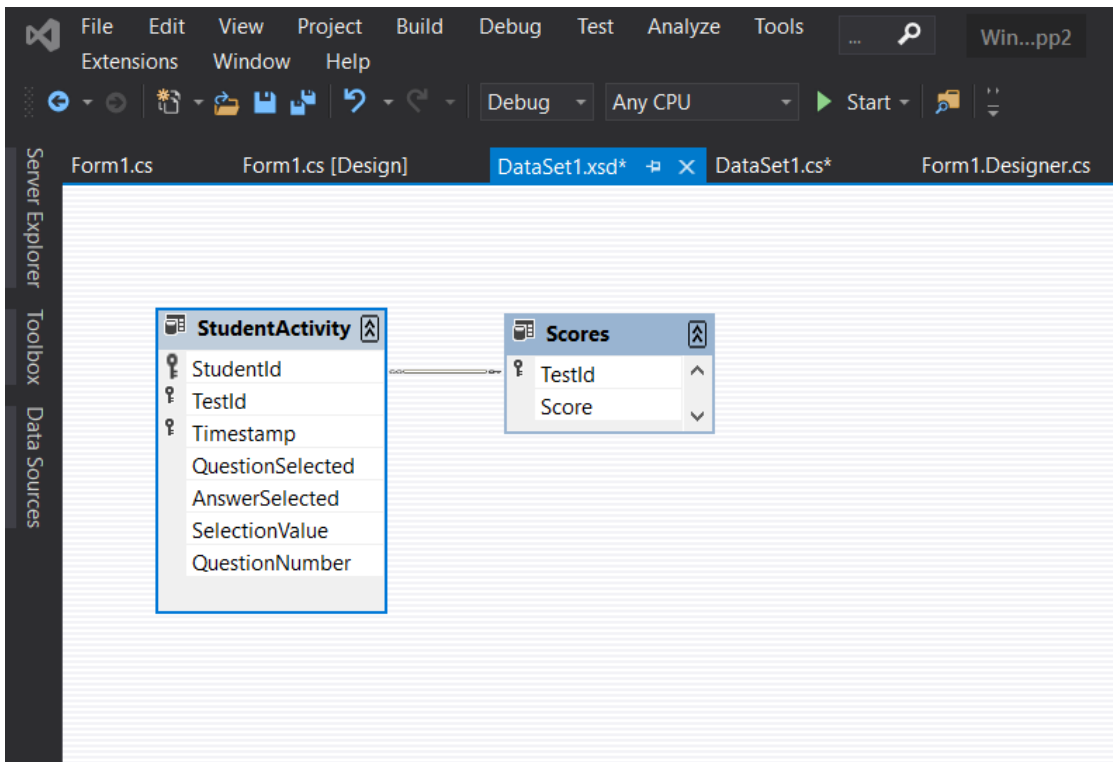
The QuestionNumber stores which question the student jumps on to next.

We have another class Scores which will calculate the total score the student gets in a paper. This score will determine what behaviour regulated the student to achieve a particular score .

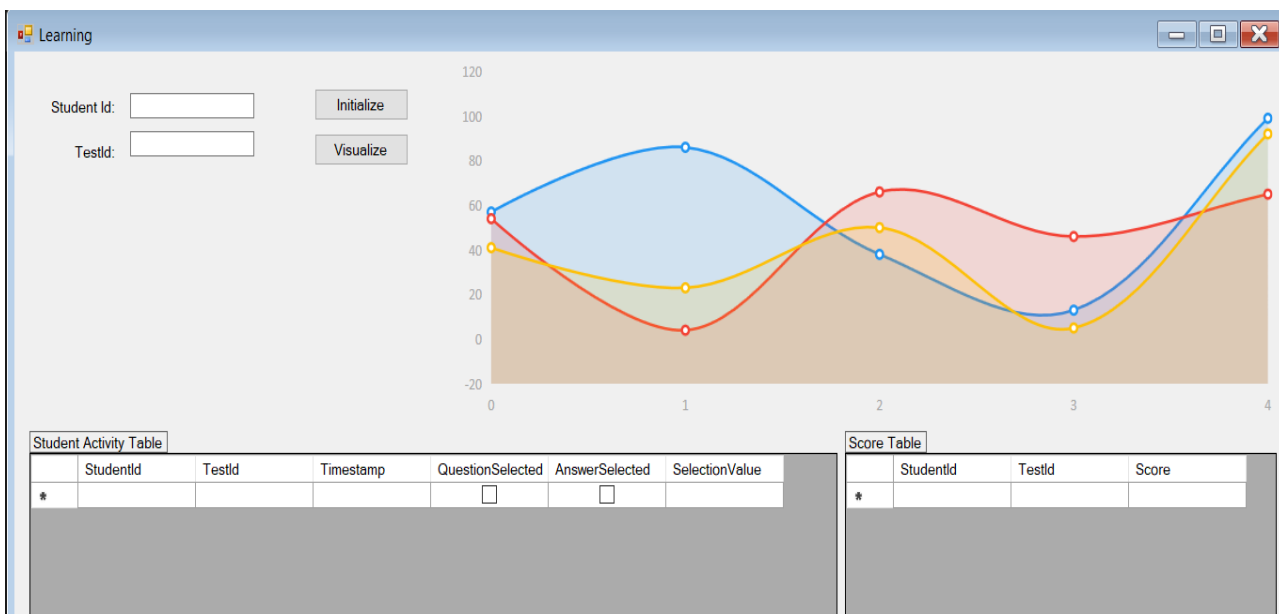
Within the score class we have three variable which are :

- StudentId
- TestId
- Score

The score of a particular student will be stored corresponding to the students Id and the Test ID of the test he/she is giving.



Next we have to form a design where in the student can enter his / her student id and test id and we can reflect upon the scores and analyse the performances versus the habit of answering in multiple choice questions.



Here we have displayed a framework where the student can input the student id and test id . When the user first clicks on initialize , the data of a student will be gathered from the sql

table which stores his/ her data stored in variables in StudentActivity class as well as score from Scores class the score of the student. When we click on visualize , the graph of the user activity is displayed .

3.5 Computational :

While we have prepared the design schema of the application , behind the background works the logic which takes into many variables like :

- Duration a student spends on each question
- Activity of intervals within a particular time
- Response given by the student
- Total time spent by the student on each question
- Frequent activity like jumping from one question to another with or without selecting the response

When a student gives a test , the scope of things which can happen are :

- A student sees a question , gives the answer to the question and moves to the next question
- A student sees the question and doesn't answer it and moves to next question
- A student while after answering more than 3-4 questions goes back to previously answered questions and looks at them again without changing them
- A student while after answering more than 3-4 questions goes back to previously answered questions and looks at them again and changing them as he goes
- A student jumps from questions in a random manner
- A student gives all the answers to the questions and then revises each question from the beginning.
- A student looks at every question and doesn't respond to any of them.

The output graph will have two axes . The X-axis will represent the timestamp at which the student does some activity like selecting which question to attempt or which answer to select or switching to another question.

The horizontal segment will be proportional to duration a person takes in answering some question or switching to another question.

The Y axes shows the question number on which some activity is being done .

The Logging framework has been applied in Form’s file where we have a framework to initialize the data and visualize the output . The form will have the data of the student which consist of the question the student select , the answers he gives , the timestamp at the time he does these activities , and the select switch to next question.

Chapter-4

Performance Analysis

4.1 Analysis:

On submitting the data of a given test by two students , I found various types of strategies adopted by a student .

Follwing strategies were discovered:

1. Linear increment:

```

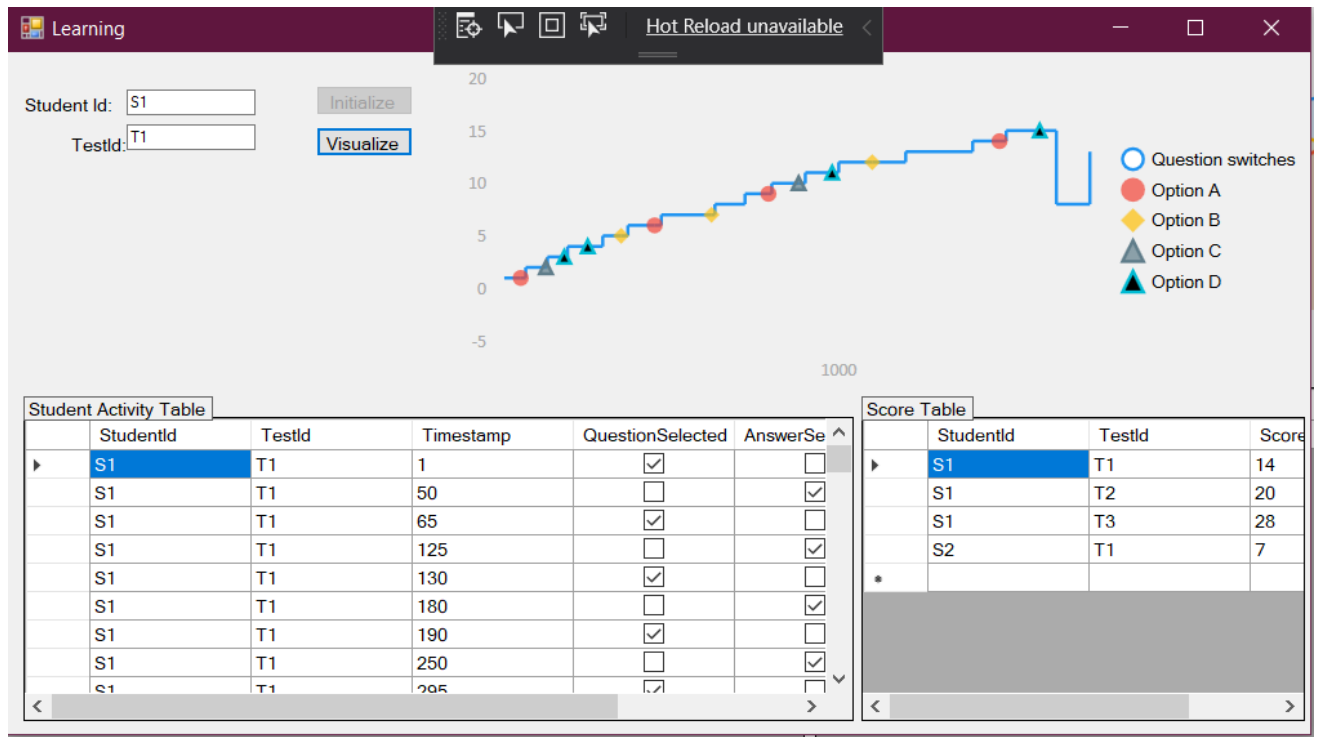
1 reference
private void initializeDataFor_S1_T1()
{
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 1, "1", 1));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 50, "a", 1));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 65, "2", 2));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 125, "c", 2));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 130, "3", 3));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 180, "d", 3));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 190, "4", 4));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 250, "d", 4));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 295, "5", 5));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 350, "b", 5));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 370, "6", 6));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 450, "a", 6));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 470, "7", 7));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 620, "b", 7));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 630, "8", 8));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 720, "9", 9));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 790, "a", 9));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 800, "10", 10));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 880, "c", 10));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 900, "11", 11));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 980, "d", 11));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 1000, "12", 12));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 1100, "b", 12));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 1200, "13", 13));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 1400, "14", 14));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 1480, "a", 14));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 1500, "15", 15));
    this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T1", 1600, "d", 15));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 1650, "8", 8));
    this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T1", 1750, "13", 13));
}

```

Assuming this test was given by Student S1 , While this strategy seemed to be simply straight forward where the student seemed to linearly keep answering the questions and linearly increment to the next question and attempting almost every question on the way and then at the end of the test did not revise his previous answers.

This strategy developed an average score of 14/30 . The possible conclusion from this derived is that the student did not keep aside the time to revise his answers and was confused in his concepts because although he attempted almost all the questions, still he did not get enough score. This reflects those category of students who lack the tendency to revise and need to be encouraged to revise their work enough times .

The graph obtained looks like this:



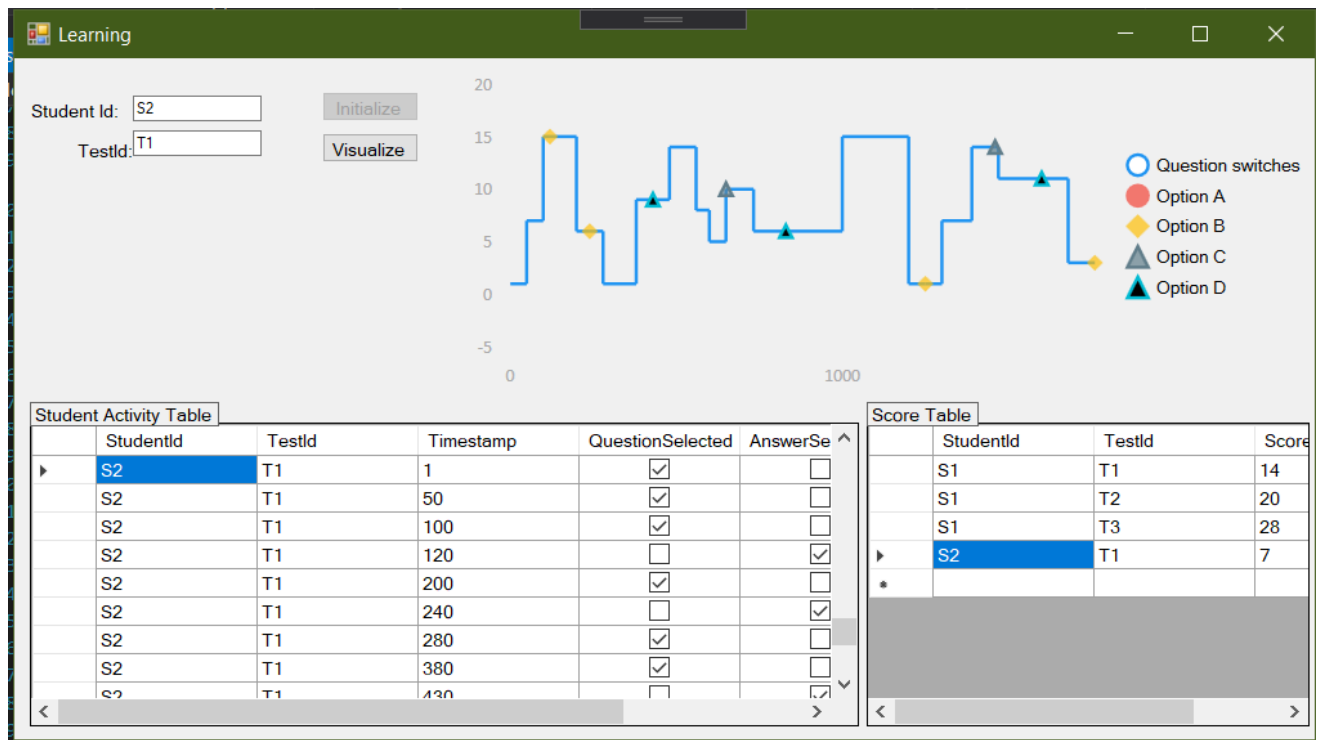
Here the graph depicts how the student S1 linearly jumps to next questions and answers them and he moves on and then at the end does not revise his responses and gets an average score out of total 30 where each question weighs 2 marks each.

When the same test was given by student S2, his strategy seemed to be very different and atypical. For Student S2, the data of variables is ;

```
297  
298  
299  
300 1 reference  
301 private void initializeDataFor_S2_T1()  
302 {  
303     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 1, "1", 1));  
304  
305     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 50, "2", 7));  
306     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 100, "15", 15));  
307     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S2", "T1", 120, "b", 15));  
308     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S2", "T1", 240, "b", 6));  
309     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 280, "1", 1));  
310     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 380, "9", 9));  
311     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S2", "T1", 430, "d", 9));  
312     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 480, "14", 14));  
313     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 560, "8", 8));  
314     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 600, "5", 5));  
315     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S2", "T1", 650, "c", 10));  
316     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 733, "6", 6));  
317     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S2", "T1", 830, "d", 6));  
318     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 1000, "15", 15));  
319     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 1200, "1", 1));  
320     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S2", "T1", 1250, "b", 1));  
321     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 1300, "7", 7));  
322     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 1390, "14", 14));  
323     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S2", "T1", 1460, "c", 14));  
324     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 1470, "11", 11));  
325     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S2", "T1", 1600, "d", 11));  
326     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S2", "T1", 1680, "9", 3));  
327     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S2", "T1", 1760, "b", 3));  
328  
329  
330 1 reference  
331 private void studentIdTextBox_TextChanged(object sender, EventArgs e)  
332 {  
333     if (this.studentIdTextBox.Text == "S1" || this.studentIdTextBox.Text == "S2")  
334     {
```

Here the student was not very responsive towards the initial questions and then switched between questions in a very haphazard manner. There was no relationship between the order of his answering the questions. He scored a very low score of 7/30 which is primarily because he didn't answer enough questions and also couldn't revise. This strategy reveals that the student wasn't prepared for the test and was baffled by the initial questions and then randomly kept switching between questions in order to finish the test without aiming for a high score.

The graph depicted as follows :



This graph shows the anomaly between answering the questions in a vague manner and the low score acquired in the result . The graph lines are very random and keep undulating with different timestamp.

When comparing the performance of student 1 and student 2 on the same test , we can see that when the student was prepared enough to atleast answer the questions , he go more than the student who kept jumping between questions . So the graph shows that both the students need to learn in different spheres.

Student 1 needs to revise his concepts and the Student 2 needs to understand the concepts well enough to answer the questions .

2.Active Revision:

```

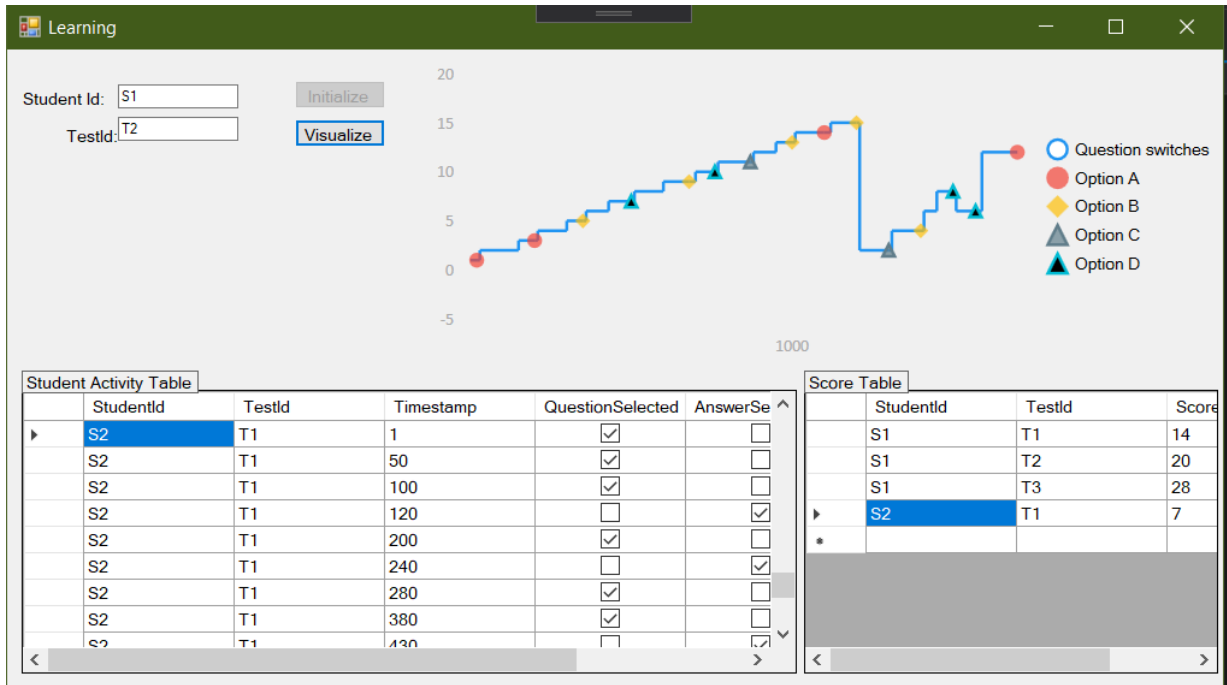
206     private void initializeDataFor_S1_T2()
207     {
208         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 1, "1", 1));
209         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 20, "a", 1));
210         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 30, "2", 2));
211         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 150, "3", 3));
212         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 200, "a", 3));
213         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 210, "4", 4));
214         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 300, "5", 5));
215         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 350, "b", 5));
216         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 360, "6", 6));
217         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 430, "7", 7));
218         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 500, "d", 7));
219         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 510, "8", 8));
220         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 600, "9", 9));
221         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 680, "b", 9));
222         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 700, "10", 10));
223         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 760, "d", 10));
224         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 770, "11", 11));
225         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 870, "c", 11));
226         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 880, "12", 12));
227         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 950, "13", 13));
228         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 1000, "b", 13));
229         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 1010, "14", 14));
230         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 1100, "a", 14));
231         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 1120, "15", 15));
232         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 1200, "b", 15));
233         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 1210, "2", 2));
234         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 1300, "c", 2));
235         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 1310, "4", 4));
236         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 1400, "b", 4));
237         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 1410, "6", 6));
238         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 1450, "8", 8));
239         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 1500, "d", 8));
240         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 1510, "6", 6));
241         this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T2", 1570, "d", 6));
242         this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T2", 1590, "12", 12));

```

This strategy was found in Student 1 giving another test T2 in which he actively answered all the questions a short phase of time and then he took time out to revise the questions and revisit those questions which he couldn't attempt and then answered those and finished his test . He had a score of 20/30 in this test .

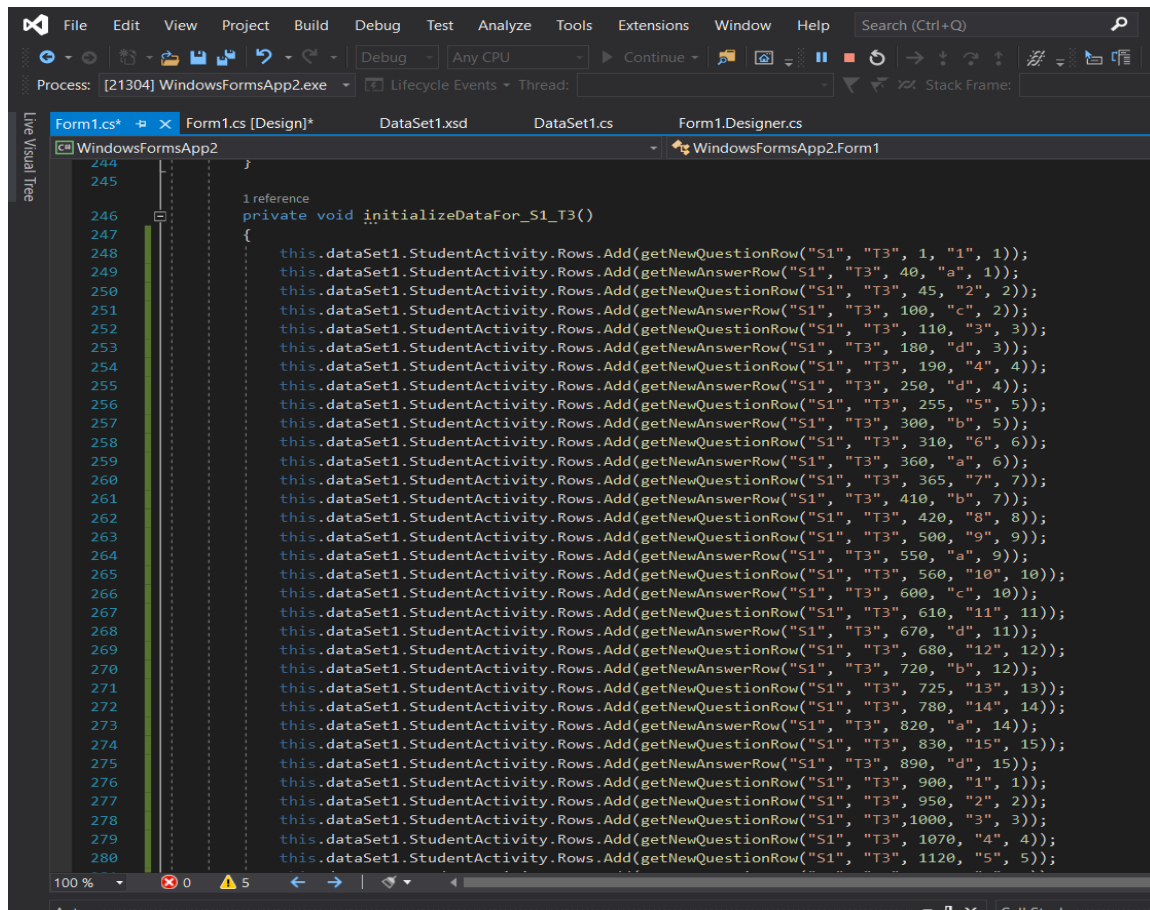
The student gave more time to attempting the questions and less time to revising and completing the unmarked questions .

The graph depicted by this test is as follows :



This graph depicts that the student spend more than half time in the first half of the test linearly jumping through questions and answering them whilst leaving the ones he didn't know and then when he reached the end , he started revising the left over questions and started solving the ones he couldn't answer before. He also revised the ones he had already did before . What is interesting to notice here is that more time is spent in the solving phase than the revising phase.

3.Passive Revising:

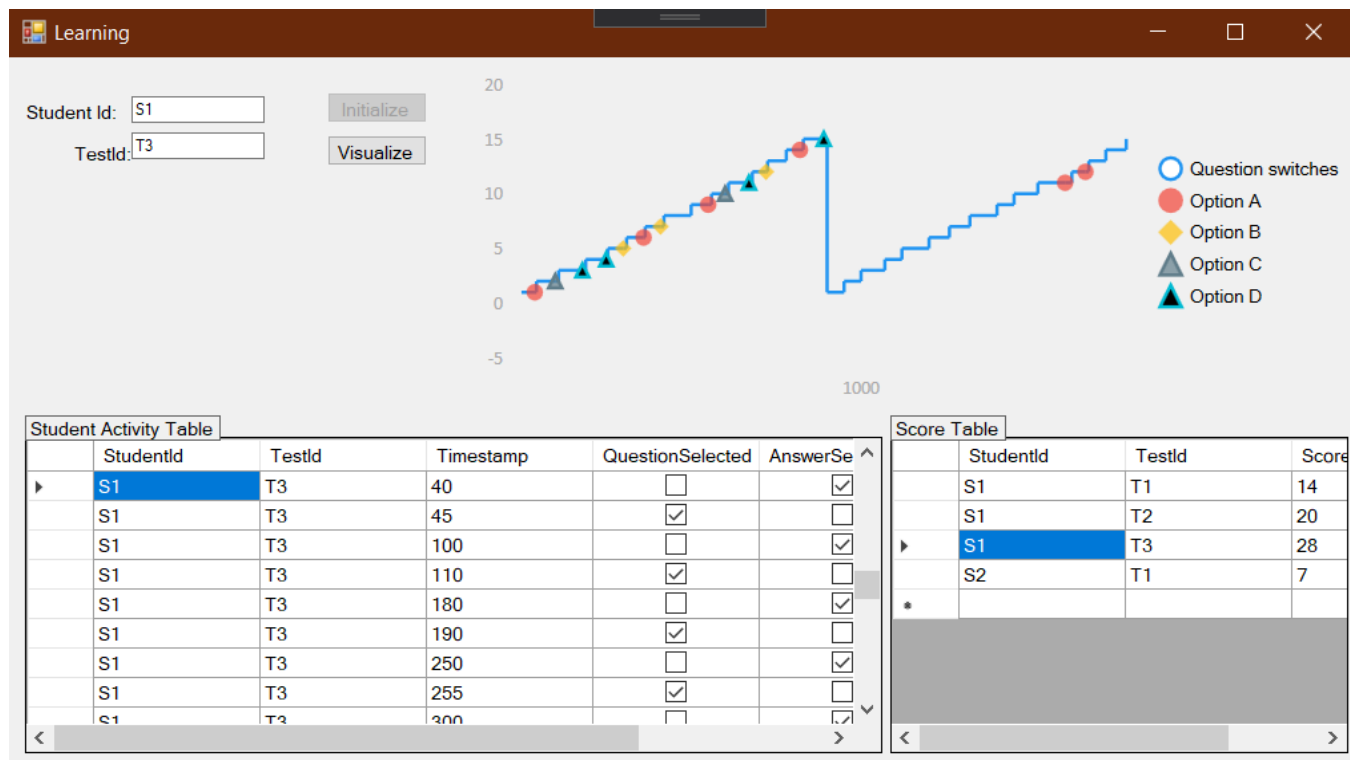


```
246 private void initializeDataFor_S1_T3()  
247 {  
248     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 1, "1", 1));  
249     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 40, "a", 1));  
250     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 45, "2", 2));  
251     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 100, "c", 2));  
252     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 110, "3", 3));  
253     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 180, "d", 3));  
254     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 190, "4", 4));  
255     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 250, "d", 4));  
256     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 255, "5", 5));  
257     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 300, "b", 5));  
258     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 310, "6", 6));  
259     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 360, "a", 6));  
260     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 365, "7", 7));  
261     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 410, "b", 7));  
262     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 420, "8", 8));  
263     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 500, "9", 9));  
264     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 550, "a", 9));  
265     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 560, "10", 10));  
266     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 600, "c", 10));  
267     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 610, "11", 11));  
268     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 670, "d", 11));  
269     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 680, "12", 12));  
270     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 720, "b", 12));  
271     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 725, "13", 13));  
272     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 780, "14", 14));  
273     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 820, "a", 14));  
274     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 830, "15", 15));  
275     this.dataSet1.StudentActivity.Rows.Add(getNewAnswerRow("S1", "T3", 890, "d", 15));  
276     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 900, "1", 1));  
277     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 950, "2", 2));  
278     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 1000, "3", 3));  
279     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 1070, "4", 4));  
280     this.dataSet1.StudentActivity.Rows.Add(getNewQuestionRow("S1", "T3", 1120, "5", 5));  
}
```

Here in this test , student gives the test with actively and firstly answering the questions and then revising them all after completing them . The student answers the questions linearly as fast as possible and then spending more than half time in passively revising the questions he has already answered. He only check the previously answered questions . This methods shows the student was actively prepared for this test because he has given enough time to complete the marking of the answers and also actively revises them once has responded ot all of them and this obtains him the highest score of 28/30 in the test.

So according to all the graphs that we have seen , the graph for this strategy achieves the highest score among all of the strategies .

The graph for this score is depicted here :



Here we can see from the graph that the student first attempt all the questions in the minimum time possible and then gives substantial amount of time revisiting all the questions again to recheck the answers. This passive technique to answering the questions shows the substantial preparation of the student before the test letting him obtain 28/30 .

When we compare this strategy with the active revision technique strategy we observe that the only difference was in revision technique . While in the former the student is spending revisiting all the questions and then submitting ,the latter sees revisiting only the ones unanswered .

From all the strategies becomes clear that the graphical pictures do correlate with the score the student has obtained in the test .

4.2 Output

In various strategies we see how the graph correlates with the score obtained by the student . When the student adopts the linear increment strategy the score is less than average because here the student does not revise his work and keeps on answering all the questions . Whereas the strategies which have more than one phase like in active revising or passive revising , the score is improved which maybe due to enhanced reasoning and getting comfortable with the test as it goes . Passive revising earns more because in active revising the student does less

reasoning when a difficult questions comes by , but in passive one the student revises them all and reasons with all types of questions . The strategy with haphazard increments and decrements states that the student has no definite clue as to how to answer the questions .

What these strategies reveal are the attitude of the student with the tests and also how much they believe in revising their work and what are their weak areas. Are the questions ther do not attempt belong to the same concept ? Are the questions being skipped not understandable to them ? Is the student completely unaware about the concepts being asked in the test ? Is the student overconfident in his skills so as not to revise their answers ? Is the student adaptive to the test where as soon as he reaches a topic , he tries going back to same topics already been asked before to recheck those answers ?

All these skills reflect upon the mental thought process that goes on in a students mind while answering the questions of a multiple choice question.

CHAPTER 5

CONCLUSION

5.1. Conclusion

An Approach has been presented to let the teachers analyse the student's learning ability by monitoring the test answers and the time taken to answer different questions. Data visualization helped us in analysing the graph of the output of the students answers. We can analyse the behavioural pattern behind answering the questions and jumping through various questions and the duration for which a student spends on a particular question. We can also study how the students patterns of revising reflects his score in the test . We can categorise it into various strategies . In this project Namely three strategies have been focussed on where a student just linearly keeps answering the questions , another where he does actively revision and yet another where the revision was mainly passive . His score were varying accordingly . We also saw an unprepared student kept jumping through random questions without answering them and ultimately a low score . This type of application can help us determine the learning pattern of a student and help the teacher to figure out where lies the problem in understanding the concepts. The question paper could be made more intuitive to the understanding of the student in order to check which concepts are more clear to the student and which are not.

5.2 Future Scope

In the future I would like to perform more experiments on various students and different types of multiple choice question tests and analyse different patterns that come our way. I would also like to insert the element of figuring out if two or more students are cheating in the test. This could be done by calling different methods and trace the questions being attempted by the students and check whether two or more students are attempting the test in the same pattern . I would like to figure out two the students can be made to understand different concept on the basis of what type of strategy they adopt in a given test . I would analyse their behavioural patterns while attempting the test especially those who are consistently achieving a low score . This project helps us to greatly impact the assessment capabilities of students by understanding their strategies and understandings especially in times of COVID-19 when online teaching will the new way and multiple choice questions will be faster and less time consuming to test how much the students are learning without their presence in the classroom.

REFERENCES

1. J. Bath, "Answer-Changing Behaviour on Objective Examinations," J. Educational Research.
2. J. Johnston, "Exam Taking Speed and Grades," Teaching of Psychology.
3. L. McClain, "Behavior During Examinations: A Comparison of 'A,' 'C,' and 'F' Students," Teaching of Psychology.
4. R. Mazza and V. Dimitrova, "Student Tracking and Personalization: Visualising Student Tracking Data to Support Instructors in Web-Based Distance Education," Proc. 13th Int'l World Wide Web Conf. Alternate Track Papers and Posters.
5. D.A. Keim and M. Ward, "Visualization," Intelligent Data Analysis, M. Berthold and D.J. Hand, eds., second ed. Springer, 2003.
6. Monitoring Online Tests through Data Visualization
Gennaro Costagliola, Member, IEEE, Vittorio Fuccella, Massimiliano Giordano, and Giuseppe
7. <http://ubiq.co/data-visualization-applications>
8. https://en.wikipedia.org/wiki/Microsoft_Visual_Studio

PLAGIARISM REPORT

udisha_report_infosys.docx

ORIGINALITY REPORT

9%

SIMILARITY INDEX

7%

INTERNET SOURCES

1%

PUBLICATIONS

7%

STUDENT PAPERS

PRIMARY SOURCES

1	www.geeksforgeeks.org Internet Source	2%
2	Submitted to Higher Education Commission Pakistan Student Paper	2%
3	www.w3schools.com Internet Source	1%
4	Submitted to Jaypee University of Information Technology Student Paper	1%
5	Submitted to International School of Management and Technology (ISMT), Nepal Student Paper	1%
6	mitpolytechnic.ac.in Internet Source	1%
7	www.scribd.com Internet Source	1%
8	sunkupuli.com Internet Source	<1%

9	Submitted to Colorado Technical University Online Student Paper	<1%
10	Submitted to Rivier University Student Paper	<1%
11	Submitted to International School of Management and Technology Student Paper	<1%
12	www.coursehero.com Internet Source	<1%
13	Michael A. Smith. "Object-Oriented Software in C++", Springer Science and Business Media LLC, 1993 Publication	<1%
14	Submitted to Kolej Universiti Linton Student Paper	<1%

Exclude quotes On

Exclude matches < 20 words

Exclude bibliography On

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 3 November,2020

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

Name: Udisha Pandey Department: IT Enrolment No 161469

Contact No. 7991827683 E-mail. udisha2810pandey@gmail.com

Name of the Supervisor: Dr. Monika Bharti

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): Assessment of Student Behaviour in class tests through Data Visualization

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 = 62
- 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** at9.....(%). 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