



TORCH BEARER CONVENT SCHOOL

(Sr. Secondary School Affiliated to CBSE)

PROJECT FILE

Subject: **Information Technology**

Class: **XII**

Academic Session: **2025–26**

Project Title:

"A Comprehensive Study of SQL Queries, Java Programs, and Web-Based Applications"

Submitted By:

Name: **Rohan Naagar**

Class: **XII – Non-Medical**

Roll No.: **11**

School Name: **Torch Bearer Convent School**

Submitted To:

Subject Teacher's Name: **Mr. Raju Gupta**

Designation: **Teacher(Information Technology)**

Board:

CENTRAL BOARD OF SECONDARY EDUCATION (CBSE) New Delhi, India



INDEX

Class 12 Project File – Academic Session 2025–26

S. No.	Content
1.	Certificate
2.	Acknowledgement
3.	SQL Queries (Practical Implementation)
4.	Java Programs (Console-Based)
5.	Operating Web-Based Applications
6.	Bibliography
7.	Appendix / Annexure

◆ SQL QUERIES

S. No.	Topic
1.	Creation of a database using MySQL command
2.	Deletion of a database using MySQL command
3.	Creation of a table using MySQL command
4.	Show the databases and tables created in MySQL
5.	List out the field name along with its field type and constraints
6.	Set a default value for a field in a table
7.	INSERT values into the table
8.	Display the details of a field based on a given condition
9.	UPDATE the details of a field based on a given condition
10.	Use of AGGREGATE FUNCTIONS in MySQL
11.	Add a new column using ALTER TABLE
12.	Delete a record from a table
13.	Sort data in ascending order using ORDER BY
14.	Count the number of students using COUNT()
15.	Rename a column using ALTER TABLE CHANGE

◆ JAVA PROGRAMS

S. No.	Topic
1.	Print the string "Hello World"
2.	Display characters A to Z using for loop
3.	Check if a number is Positive, Negative, or Zero using if...else
4.	Find the largest among three numbers using if...else
5.	Check whether a number is even or odd using if...else
6.	Generate multiplication table (1 to 10) using for loop
7.	Find the area of a rectangle by taking length and breadth as input
8.	Print the sum of the first 10 natural numbers
9.	Generate Fibonacci series based on user input
10.	Print the factorial of a given number
11.	Calculate the sum of digits of a number
12.	Check whether a number is prime
13.	Reverse a number
14.	Check whether a string is a palindrome
15.	Count the number of vowels in a string

◆ OPERATING WEB-BASED APPLICATIONS

S. No.	Topic
1.	Case Study – Online Bill Calculator / Book Rail Ticket
2.	Case Study – Online Quiz: Internet and Phone Safety for School Children
3.	Case Study – Online Game: Hangman

CERTIFICATE

This is to certify that **Rohan Naagar**, a student of **Class XII – Non-Medical** , has successfully completed the project work titled:

“A Comprehensive Study of SQL Queries, Java Programs, and Web-Based Applications”

as per the guidelines issued by the **Central Board of Secondary Education (CBSE)**, for the academic session **2025–26**, under the subject **Information Technology (code-802)**.

The project submitted is an original work undertaken by the student and has been completed under my supervision and guidance.

I wish him/her all the best for future academic endeavors.

Signature of the Subject Teacher

Name: _____

Designation: _____

Date: _____

Signature of the Principal

Name: _____

School Seal:

Date: _____

Student Details:

Name: **Rohan Naagar**

Class & Section: **XII – Non-Medical**

Roll Number: **11**

Subject: **Information Technology**



Scan the QR for profile

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who helped me complete this project successfully.

First and foremost, I am thankful to the **Almighty** for providing me the strength and determination to undertake and complete this project.

I extend my heartfelt thanks to my **subject teacher, Mr. Raju Gupta**, for their constant support, valuable guidance, and encouragement throughout the course of this project. Their insightful suggestions and constructive feedback played a vital role in shaping the final outcome.

I am also grateful to our **Principal, Mrs. Anjali Kaushik**, for providing the necessary infrastructure and academic environment to carry out this work.

I would also like to thank my **parents and friends** for their moral support, patience, and cooperation during the preparation of this project.

Finally, I acknowledge that this project is the result of dedication, discipline, and collaborative effort, and I am proud to submit it as a part of my Class 12 curriculum.

Name: Rohan Naagar

Class: XII – Non-Medical

Roll Number: 11

Subject: Information Technology

SQL QUERIES

1. Write down syntax and MySQL program to create a database STUDENT and its OUTPUT.

AIM

To write a MySQL code to create a database STUDENT, listing out its **SYNTAX**, **PROGRAM**, and the resulting **OUTPUT**.

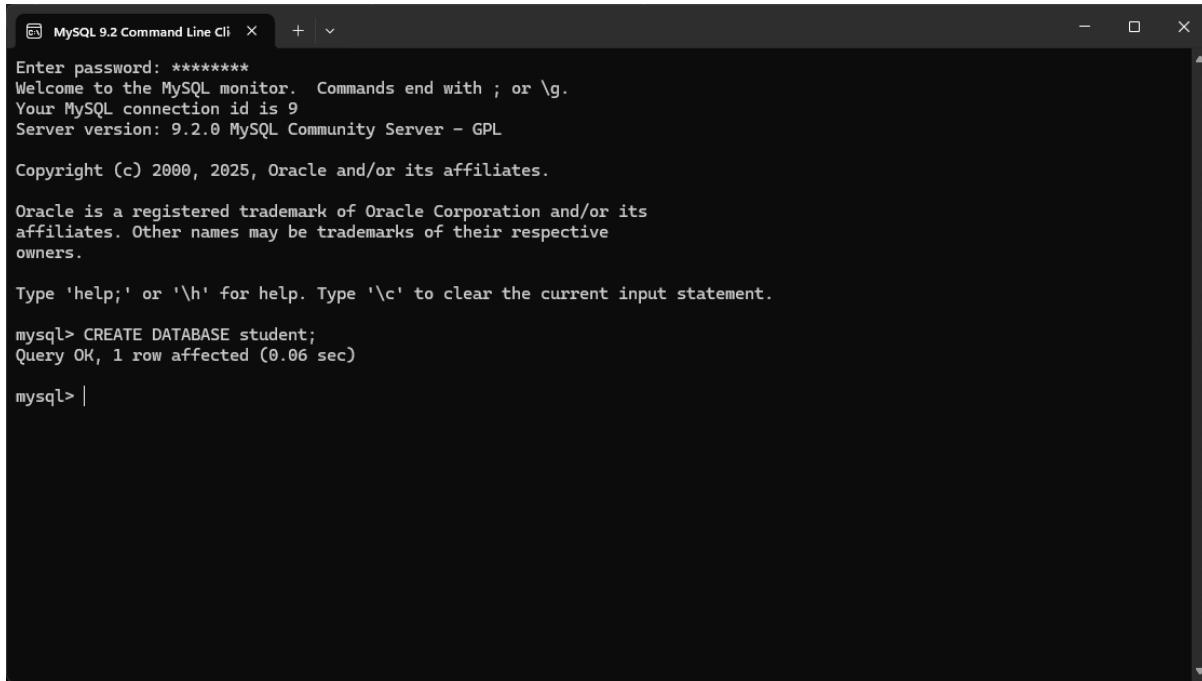
SYNTAX

CREATE DATABASE dbname;

PROGRAM

mysql> CREATE DATABASE student;

OUTPUT



The screenshot shows a terminal window titled "MySQL 9.2 Command Line Cli". It displays the MySQL monitor welcome message, including the connection id (9), server version (9.2.0), and copyright information. The user then enters the command "CREATE DATABASE student;" followed by a semicolon. The response "Query OK, 1 row affected (0.06 sec)" is shown, indicating the database was successfully created.

```
MySQL 9.2 Command Line Cli  +  -  X
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE student;
Query OK, 1 row affected (0.06 sec)

mysql> |
```

2. Write down syntax and MySQL program to delete a database STUDENT and its OUTPUT.

AIM

To write a MySQL code to delete a database STUDENT, listing out its **SYNTAX**, **PROGRAM**, and the resulting **OUTPUT**.

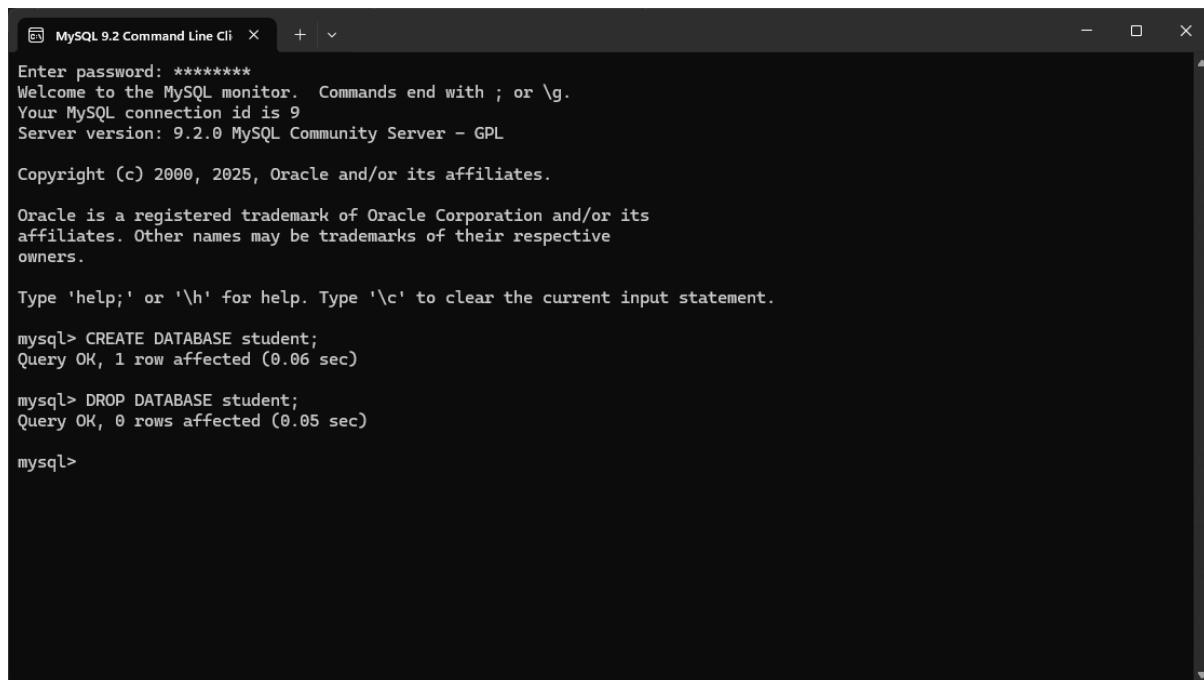
SYNTAX

DROP DATABASE dbname;

PROGRAM

mysql> DROP DATABASE student;

OUTPUT



The screenshot shows a terminal window titled "MySQL 9.2 Command Line Cli". The session starts with the MySQL prompt "mysql>". The user creates a database named "student" with the command "CREATE DATABASE student;". The response "Query OK, 1 row affected (0.06 sec)" indicates success. Later, the user drops the same database with the command "DROP DATABASE student;". The response "Query OK, 0 rows affected (0.05 sec)" indicates success. The MySQL monitor displays standard welcome messages and copyright information.

```
mysql> CREATE DATABASE student;
Query OK, 1 row affected (0.06 sec)

mysql> DROP DATABASE student;
Query OK, 0 rows affected (0.05 sec)
```

3. Write down syntax and MySQL program to create a table STUDENT and its OUTPUT.

AIM

To write a MySQL code to create a table STUDENT with fields and constraints, listing out the **SYNTAX, PROGRAM**, and the **OUTPUT**.

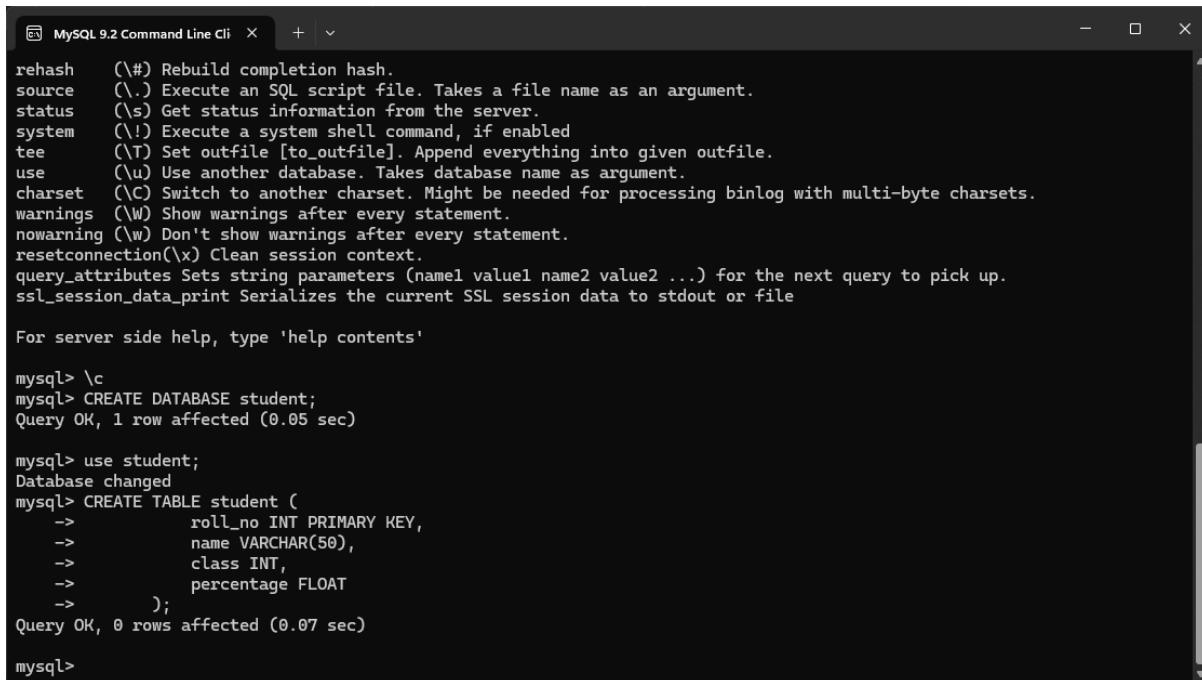
SYNTAX

```
CREATE TABLE table_name (
    column1 datatype constraint,
    column2 datatype constraint,
    ...
);
```

PROGRAM

```
mysql> CREATE TABLE student (
    roll_no INT PRIMARY KEY,
    name VARCHAR(50), class INT , percentage FLOAT );
```

OUTPUT



The screenshot shows a terminal window titled "MySQL 9.2 Command Line Cli". It displays the MySQL command-line interface. The user has run several commands to create a database and a table:

```
rehash      (\#) Rebuild completion hash.
source      (\.) Execute an SQL script file. Takes a file name as an argument.
status      (\s) Get status information from the server.
system      (\!) Execute a system shell command, if enabled
tee         (\T) Set outfile [to_outfile]. Append everything into given outfile.
use         (\u) Use another database. Takes database name as argument.
charset     (\C) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.
warnings   (\W) Show warnings after every statement.
nowarning  (\w) Don't show warnings after every statement.
resetconnection(\x) Clean session context.
query_attributes Sets string parameters (name1 value1 name2 value2 ...) for the next query to pick up.
ssl_session_data_print Serializes the current SSL session data to stdout or file

For server side help, type 'help contents'

mysql> \c
mysql> CREATE DATABASE student;
Query OK, 1 row affected (0.05 sec)

mysql> use student;
Database changed
mysql> CREATE TABLE student (
    >     roll_no INT PRIMARY KEY,
    >     name VARCHAR(50),
    >     class INT,
    >     percentage FLOAT
    > );
Query OK, 0 rows affected (0.07 sec)

mysql>
```

4. Write down syntax and MySQL program to show the databases and tables created in MySQL and its OUTPUT.

AIM

To write MySQL commands that display the **list of databases** and **tables** created, along with the **SYNTAX, PROGRAM**, and **OUTPUT**.

SYNTAX

SHOW DATABASES;

SHOW TABLES;

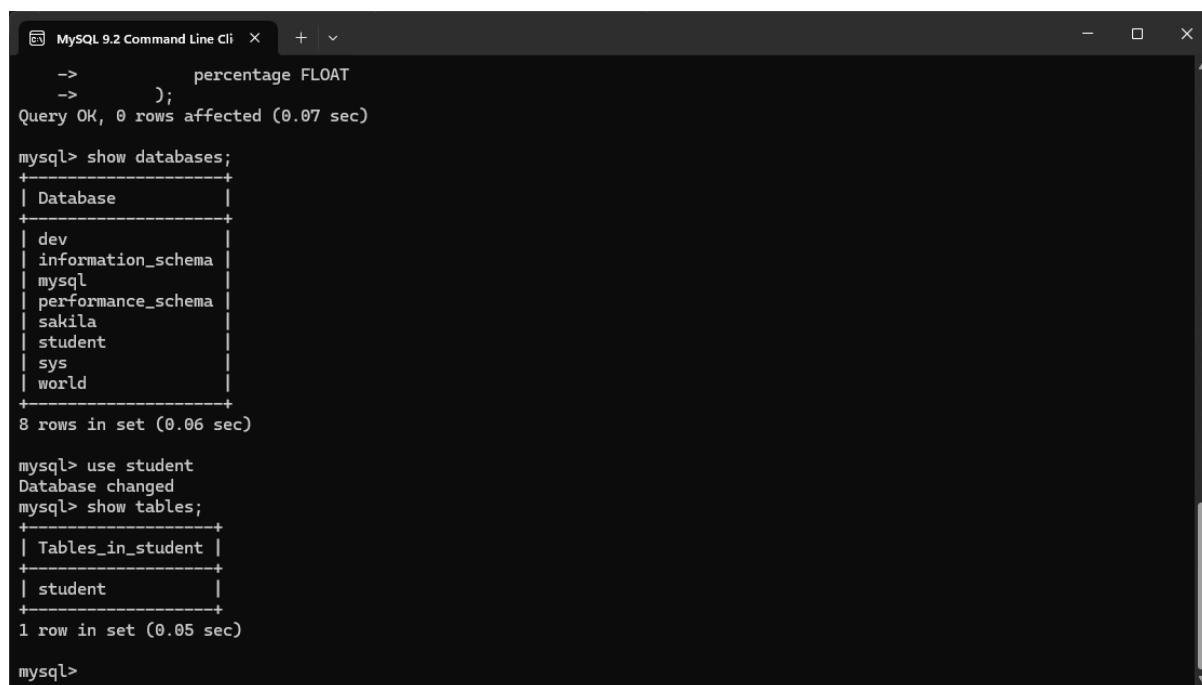
PROGRAM

mysql> SHOW DATABASES;

mysql> USE student;

mysql> SHOW TABLES;

OUTPUT



The screenshot shows a terminal window titled "MySQL 9.2 Command Line Cli". The user has run several MySQL commands:

- "SHOW DATABASES;" command, which lists the databases: dev, information_schema, mysql, performance_schema, sakila, student, sys, and world.
- "USE student;" command, changing the current database to "student".
- "SHOW TABLES;" command, which lists the tables in the "student" database: student and tables_in_student.

The output is displayed in a standard MySQL tabular format with column headers and data rows.

5. Write down syntax and MySQL program to list field names with their data types and constraints for a table, and its OUTPUT.

AIM

To write MySQL code that displays the **structure of a table** STUDENT, showing each **field name**, its **data type**, and any **constraints** applied.

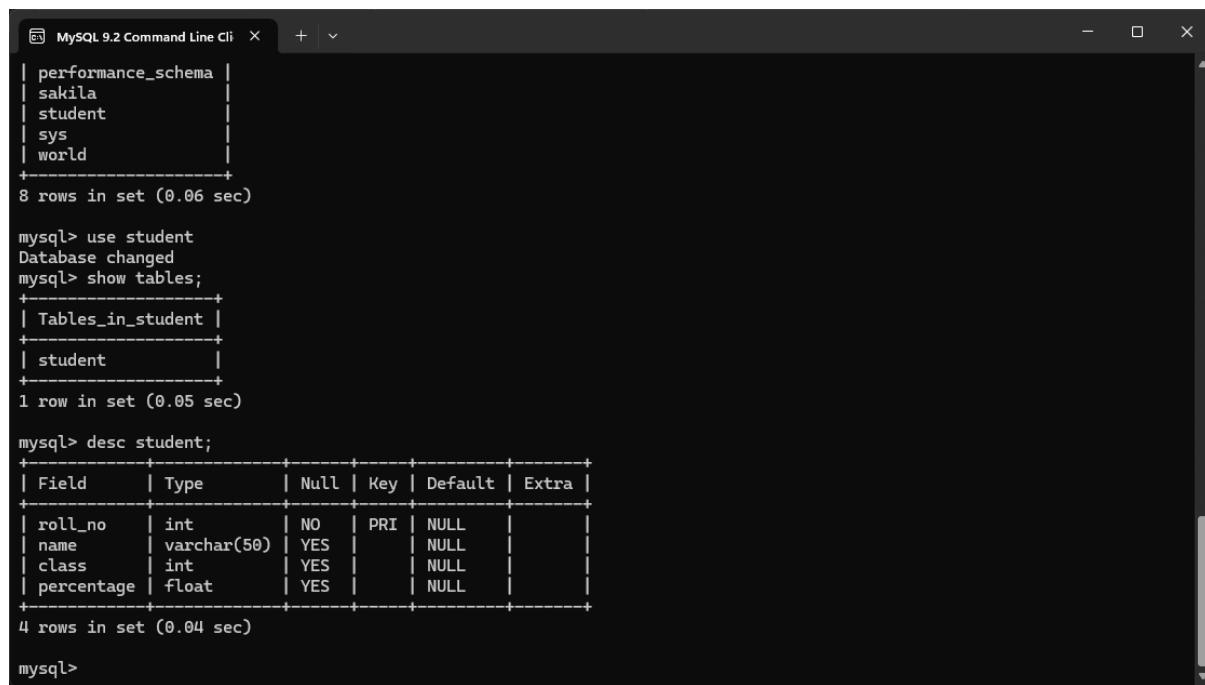
SYNTAX

```
DESC table_name;
```

PROGRAM

```
mysql> DESC student;
```

OUTPUT



```
MySQL 9.2 Command Line Cli  +  ×
+-----+
| performance_schema |
| sakila             |
| student            |
| sys                |
| world              |
+-----+
8 rows in set (0.06 sec)

mysql> use student
Database changed
mysql> show tables;
+-----+
| Tables_in_student |
+-----+
| student           |
+-----+
1 row in set (0.05 sec)

mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field    | Type     | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| roll_no  | int      | NO   | PRI | NULL    |       |
| name     | varchar(50) | YES  |     | NULL    |       |
| class    | int      | YES  |     | NULL    |       |
| percentage | float   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.04 sec)

mysql>
```

6. Write down syntax and MySQL program to set a default value for a field in a table and its OUTPUT.

AIM

To write MySQL code that sets a **default value** for a field in the table STUDENT, showing **SYNTAX, PROGRAM, and OUTPUT**.

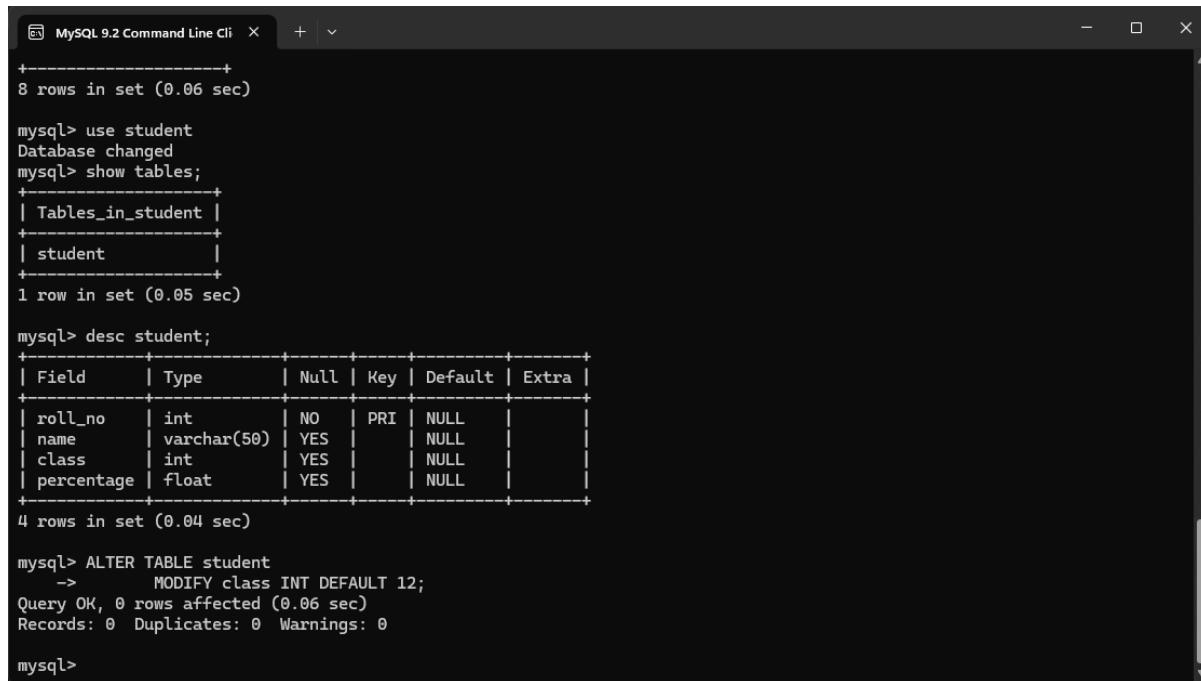
SYNTAX

```
ALTER TABLE table_name  
MODIFY column_name datatype DEFAULT default_value;
```

PROGRAM

```
mysql> ALTER TABLE student  
        MODIFY class INT DEFAULT 12;
```

OUTPUT



The screenshot shows a terminal window titled "MySQL 9.2 Command Line Cli". The session starts with:

```
+-----+  
8 rows in set (0.06 sec)
```

Then the user runs:

```
mysql> use student  
Database changed
```

Shows tables:

```
+-----+  
| Tables_in_student |  
+-----+  
| student           |  
+-----+
```

Shows the student table structure:

```
1 row in set (0.05 sec)
```

```
mysql> desc student;  
+-----+-----+-----+-----+-----+  
| Field    | Type     | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
| roll_no  | int      | NO   | PRI | NULL    |       |  
| name     | varchar(50) | YES  |     | NULL    |       |  
| class    | int      | YES  |     | NULL    |       |  
| percentage | float   | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+
```

Shows the result of the ALTER TABLE command:

```
4 rows in set (0.04 sec)
```

```
mysql> ALTER TABLE student  
        ->     MODIFY class INT DEFAULT 12;  
Query OK, 0 rows affected (0.06 sec)  
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql>
```

7. Write down syntax and MySQL program to insert values into a table and its OUTPUT.

AIM

To write MySQL code to **insert records** into the STUDENT table, showing **SYNTAX, PROGRAM, and OUTPUT**.

SYNTAX

INSERT INTO table_name (column1, column2, ...)

VALUES (value1, value2, ...);

PROGRAM

```
mysql> INSERT INTO student (roll_no, name, class, percentage)
```

```
VALUES (101, 'Rohan', 12, 91.5);
```

OUTPUT

```
MySQL 9.2 Command Line Cli + - X
Database changed
mysql> show tables;
+-----+
| Tables_in_student |
+-----+
| student           |
+-----+
1 row in set (0.05 sec)

mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field    | Type     | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| roll_no  | int      | NO   | PRI | NULL    |       |
| name     | varchar(50) | YES  |     | NULL    |       |
| class    | int      | YES  |     | NULL    |       |
| percentage | float   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.04 sec)

mysql> ALTER TABLE student
->     MODIFY class INT DEFAULT 12;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> INSERT INTO student (roll_no, name, class, percentage)
->     VALUES (101, 'Rohan', 12, 91.5);
Query OK, 1 row affected (0.04 sec)

mysql>
```

8. Write down syntax and MySQL program to display the details of a field in a table based on a given condition and its OUTPUT.

AIM

To write a MySQL command to **retrieve specific records** from the STUDENT table using a **conditional query**, showing **SYNTAX, PROGRAM, and OUTPUT**.

SYNTAX

```
SELECT * FROM table_name
```

```
WHERE condition;
```

PROGRAM

```
mysql> SELECT * FROM student
```

```
    WHERE percentage > 90;
```

OUTPUT

The screenshot shows a terminal window titled "MySQL 9.2 Command Line Cli". The session starts with the command "desc student", which displays the structure of the "student" table with four columns: roll_no, name, class, and percentage. The "percentage" column is defined as a float type with a default value of 12. Following this, an "ALTER TABLE" command is run to change the default value of the "class" column to 12. Then, an "INSERT INTO" command is used to add a new row with values (101, 'Rohan', 12, 91.5). Finally, a "SELECT * FROM student WHERE percentage > 90;" query is executed, returning a single row for the student with roll_no 101, name Rohan, class 12, and percentage 91.5.

```
mysql> desc student;
+-----+-----+-----+-----+
| Field | Type  | Null | Key  |
+-----+-----+-----+-----+
| roll_no | int   | NO   | PRI  |
| name   | varchar(50) | YES  |       |
| class  | int   | YES  |       |
| percentage | float | YES  |       |
+-----+-----+-----+-----+
4 rows in set (0.04 sec)

mysql> ALTER TABLE student
    ->      MODIFY class INT DEFAULT 12;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> INSERT INTO student (roll_no, name, class, percentage)
    ->      VALUES (101, 'Rohan', 12, 91.5);
Query OK, 1 row affected (0.04 sec)

mysql> SELECT * FROM student
    ->      WHERE percentage > 90;
+-----+-----+-----+-----+
| roll_no | name | class | percentage |
+-----+-----+-----+-----+
|     101 | Rohan |     12 |        91.5 |
+-----+-----+-----+-----+
1 row in set (0.04 sec)

mysql>
```

9. Write down syntax and MySQL program to update the details of a field in a table based on a given condition and its OUTPUT.

AIM

To write a MySQL command to **update data** in the STUDENT table based on a specified condition, showing **SYNTAX**, **PROGRAM**, and **OUTPUT**.

SYNTAX

UPDATE table_name

SET column_name = new_value

WHERE condition;

PROGRAM

```
mysql> UPDATE student
```

```
    SET percentage = 95.0
```

```
    WHERE roll_no = 101;
```

OUTPUT

The screenshot shows a terminal window titled "MySQL 9.2 Command Line Cli". The session starts with a SELECT query to show the current values of class and percentage for all rows:

```
| class      | int          | YES   |      | NULL    |      |
| percentage | float        | YES   |      | NULL    |      |
+-----+-----+-----+-----+-----+
4 rows in set (0.04 sec)
```

Then, an ALTER TABLE command is run to change the default value of the class column to 12:

```
mysql> ALTER TABLE student
      >     MODIFY class INT DEFAULT 12;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Next, an INSERT INTO command is used to add a new row for student ID 101, name Rohan, class 12, and percentage 91.5:

```
mysql> INSERT INTO student (roll_no, name, class, percentage)
      >     VALUES (101, 'Rohan', 12, 91.5);
Query OK, 1 row affected (0.04 sec)
```

A SELECT * FROM student query is then run to verify the data:

```
mysql> SELECT * FROM student
      >     WHERE percentage > 90;
+-----+-----+-----+-----+
| roll_no | name  | class | percentage |
+-----+-----+-----+-----+
|    101  | Rohan |    12 |      91.5 |
+-----+-----+-----+-----+
1 row in set (0.04 sec)
```

Finally, an UPDATE command is executed to change the percentage of the student with ID 101 to 95.0:

```
mysql> UPDATE student
      >     SET percentage = 95.0
      >     WHERE roll_no = 101;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql>
```

10. Write down syntax and MySQL program to use aggregate functions in MySQL and its OUTPUT.

AIM

To demonstrate the use of **aggregate functions** in MySQL such as COUNT(), AVG(), MAX(), MIN(), and SUM() on the STUDENT table, showing **SYNTAX**, **PROGRAM**, and **OUTPUT**.

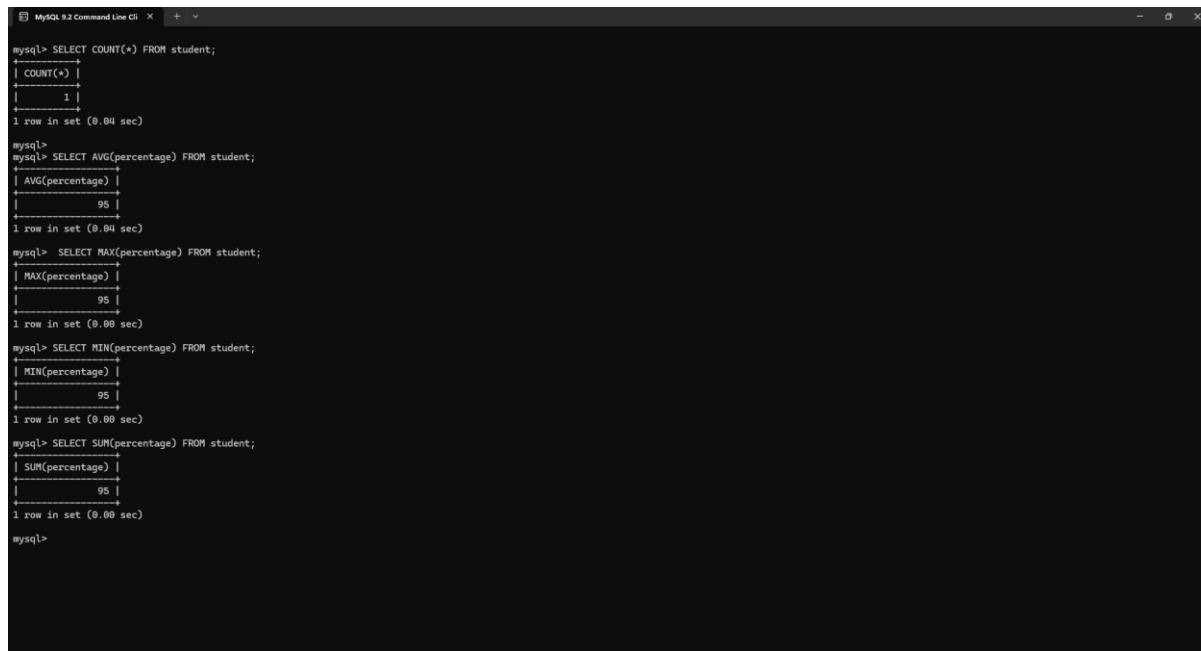
SYNTAX

```
SELECT AGGREGATE_FUNCTION(column_name)
FROM table_name;
```

PROGRAM

```
mysql> SELECT COUNT(*) FROM student;
mysql> SELECT AVG(percentage) FROM student;
mysql> SELECT MAX(percentage) FROM student;
mysql> SELECT MIN(percentage) FROM student;
mysql> SELECT SUM(percentage) FROM student;
```

OUTPUT



The screenshot shows a terminal window titled "MySQL 8.0 Command Line Cli". It displays the following SQL queries and their results:

- mysql> SELECT COUNT(*) FROM student;
| COUNT(*) |
| 1 |
1 row in set (0.00 sec)
- mysql> SELECT AVG(percentage) FROM student;
| AVG(percentage) |
| 95 |
1 row in set (0.00 sec)
- mysql> SELECT MAX(percentage) FROM student;
| MAX(percentage) |
| 95 |
1 row in set (0.00 sec)
- mysql> SELECT MIN(percentage) FROM student;
| MIN(percentage) |
| 95 |
1 row in set (0.00 sec)
- mysql> SELECT SUM(percentage) FROM student;
| SUM(percentage) |
| 95 |
1 row in set (0.00 sec)

mysql>

11. Write SQL syntax and program to add a new column to a table.

AIM

To add a new column gender to an existing table student using ALTER TABLE.

SYNTAX

ALTER TABLE table_name

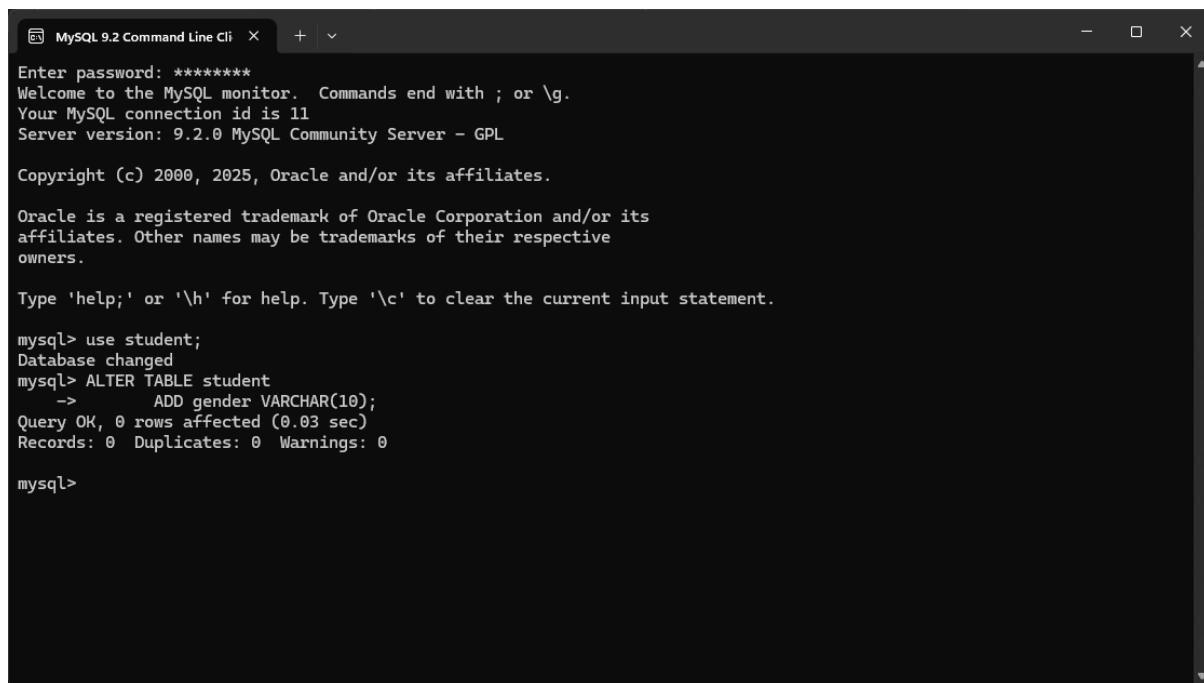
ADD column_name datatype;

PROGRAM

mysql> ALTER TABLE student

ADD gender VARCHAR(10);

OUTPUT



```
MySQL 9.2 Command Line Cli  +  ▾
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use student;
Database changed
mysql> ALTER TABLE student
    ->     ADD gender VARCHAR(10);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

12. Write SQL syntax and program to delete a specific record from a table.

AIM

To remove a particular record using the DELETE command based on a condition.

SYNTAX

DELETE FROM table_name

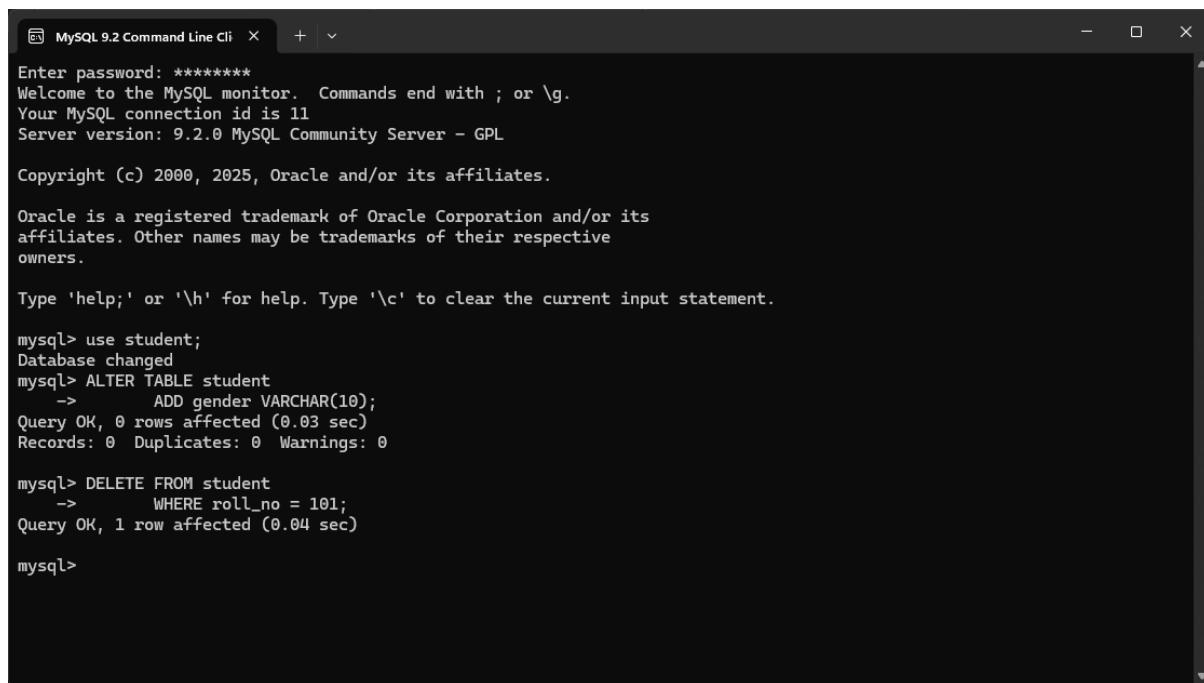
WHERE condition;

PROGRAM

mysql> DELETE FROM student

WHERE roll_no = 101;

OUTPUT



The screenshot shows a terminal window titled "MySQL 9.2 Command Line Cli". It displays the MySQL monitor welcome message, including the server version (9.2.0 MySQL Community Server - GPL). The user has entered a password. The command "use student;" is run, changing the database to "student". An "ALTER TABLE student" command is issued with an "ADD gender VARCHAR(10);". The "DELETE FROM student WHERE roll_no = 101;" command is then run, resulting in 1 row affected. The MySQL prompt "mysql>" is visible at the bottom.

```
MySQL 9.2 Command Line Cli  +  v
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use student;
Database changed
mysql> ALTER TABLE student
    ->     ADD gender VARCHAR(10);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DELETE FROM student
    ->     WHERE roll_no = 101;
Query OK, 1 row affected (0.04 sec)

mysql>
```

13. Write SQL syntax and program to sort data in ascending order.

AIM

To sort the data of a table in **ascending order** using ORDER BY.

SYNTAX

```
SELECT * FROM table_name
```

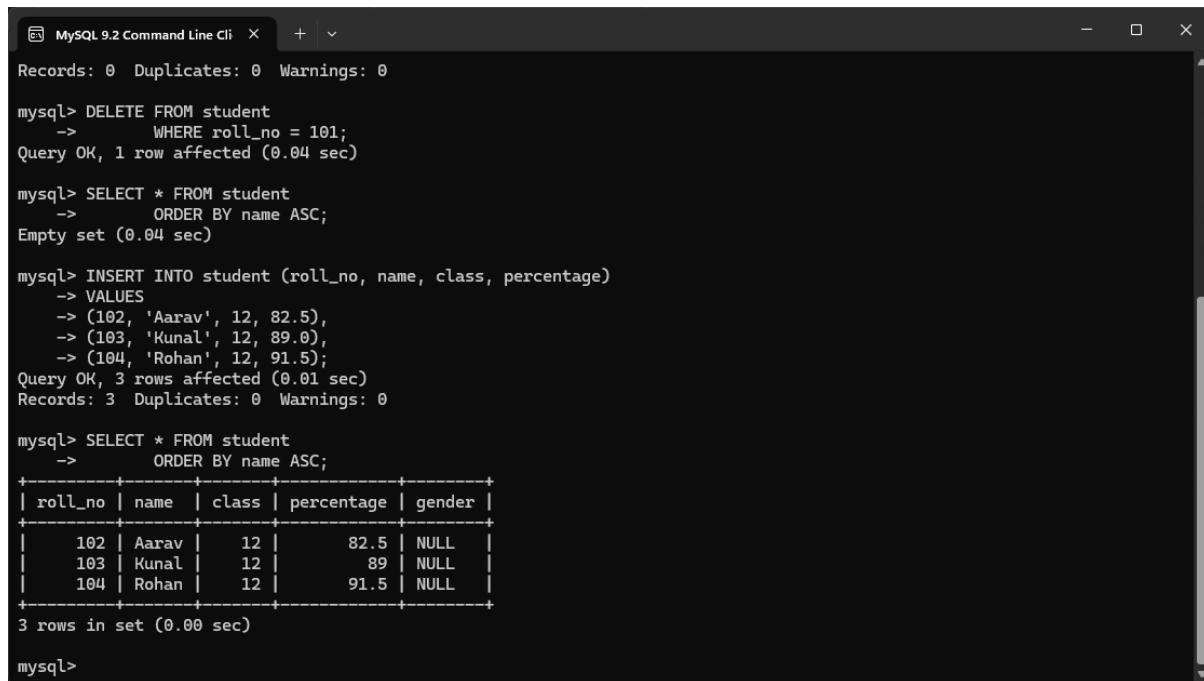
```
ORDER BY column_name ASC;
```

PROGRAM

```
mysql> SELECT * FROM student
```

```
    ORDER BY name ASC;
```

OUTPUT



The screenshot shows a terminal window titled "MySQL 9.2 Command Line Cli". The session starts with a message about records, duplicates, and warnings. Then, a DELETE query is run to remove a row where roll_no = 101. This is followed by a SELECT query that retrieves all columns from the student table and orders them by name in ascending order. The result set is empty. Next, an INSERT query is executed, inserting three new rows with roll_no values 102, 103, and 104, names Aarav, Kunal, and Rohan respectively, and class 12. The percentage values are 82.5, 89.0, and 91.5. Finally, another SELECT query is run, identical to the previous one, which now returns the three newly inserted rows in ascending order of name.

```
Records: 0  Duplicates: 0  Warnings: 0

mysql> DELETE FROM student
->      WHERE roll_no = 101;
Query OK, 1 row affected (0.04 sec)

mysql> SELECT * FROM student
->      ORDER BY name ASC;
Empty set (0.04 sec)

mysql> INSERT INTO student (roll_no, name, class, percentage)
-> VALUES
-> (102, 'Aarav', 12, 82.5),
-> (103, 'Kunal', 12, 89.0),
-> (104, 'Rohan', 12, 91.5);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM student
->      ORDER BY name ASC;
+-----+-----+-----+-----+
| roll_no | name   | class | percentage | gender |
+-----+-----+-----+-----+
|     102 | Aarav  |    12 |      82.5 |    NULL |
|     103 | Kunal   |    12 |      89.0 |    NULL |
|     104 | Rohan  |    12 |      91.5 |    NULL |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

14. Write SQL syntax and program to count the number of students.

AIM

To count the total number of records in the student table using the COUNT() aggregate function.

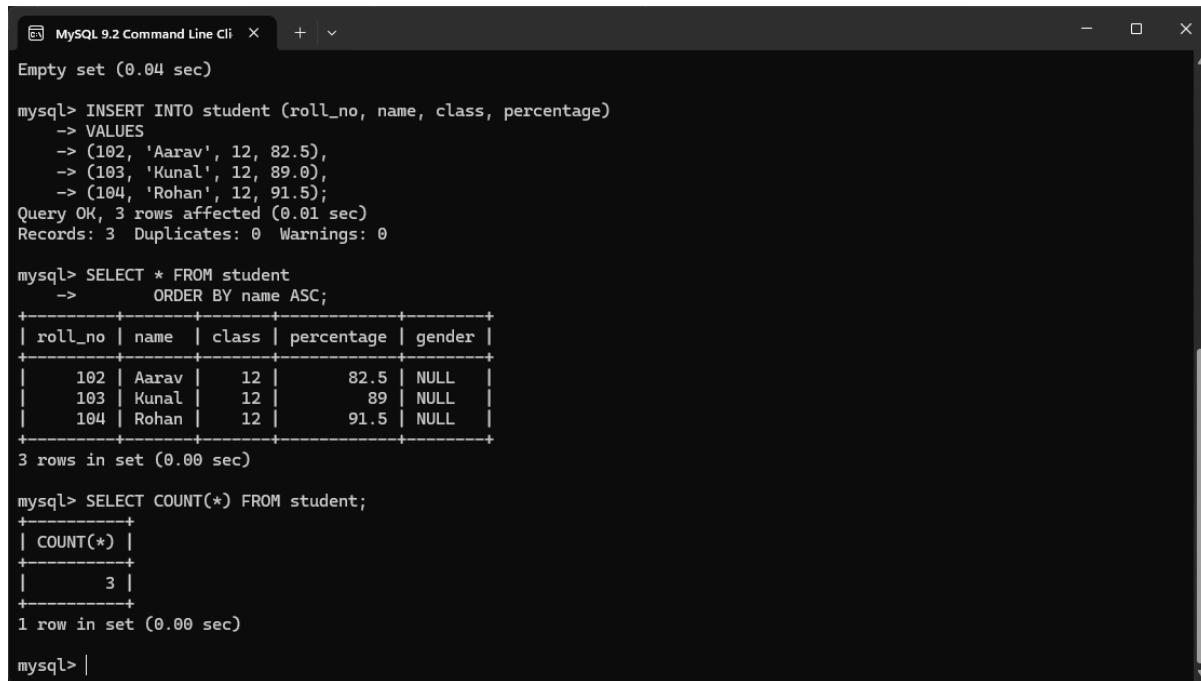
SYNTAX

```
SELECT COUNT(*) FROM table_name;
```

PROGRAM

```
mysql> SELECT COUNT(*) FROM student;
```

OUTPUT



The screenshot shows a terminal window titled "MySQL 9.2 Command Line Cli". The session starts with an empty set message, followed by an INSERT INTO command inserting three rows into the student table. The table has columns: roll_no, name, class, percentage, and gender. The inserted data is: (102, 'Aarav', 12, 82.5), (103, 'Kunal', 12, 89.0), and (104, 'Rohan', 12, 91.5). A SELECT * query is run to display all rows from the student table, ordered by name ascending. Finally, a COUNT(*) query is executed to return the total number of rows in the student table, which is 3.

```
Empty set (0.04 sec)

mysql> INSERT INTO student (roll_no, name, class, percentage)
-> VALUES
-> (102, 'Aarav', 12, 82.5),
-> (103, 'Kunal', 12, 89.0),
-> (104, 'Rohan', 12, 91.5);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM student
-> ORDER BY name ASC;
+-----+-----+-----+-----+
| roll_no | name   | class  | percentage |
+-----+-----+-----+-----+
|    102  | Aarav  |    12  |      82.5 |
|    103  | Kunal   |    12  |      89.0  |
|    104  | Rohan  |    12  |      91.5  |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT COUNT(*) FROM student;
+-----+
| COUNT(*) |
+-----+
|      3   |
+-----+
1 row in set (0.00 sec)

mysql> |
```

15. Write SQL syntax and program to rename a column in a table.

AIM

To rename the column name to full_name in the student table using ALTER TABLE.

SYNTAX

ALTER TABLE table_name

CHANGE old_column_name new_column_name datatype;

PROGRAM

mysql> ALTER TABLE student

CHANGE name full_name VARCHAR(50);

OUTPUT

```
MySQL 9.2 Command Line Cli  +  v
-> (103, 'Kunal', 12, 89.0),
-> (104, 'Rohan', 12, 91.5);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM student
->      ORDER BY name ASC;
+-----+-----+-----+-----+
| roll_no | name   | class | percentage | gender |
+-----+-----+-----+-----+
|    102 | Aarav  |    12 |      82.5 | NULL   |
|    103 | Kunal   |    12 |       89 | NULL   |
|    104 | Rohan  |    12 |      91.5 | NULL   |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT COUNT(*) FROM student;
+-----+
| COUNT(*) |
+-----+
|      3 |
+-----+
1 row in set (0.00 sec)

mysql> ALTER TABLE student
->      CHANGE name full_name VARCHAR(50);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

JAVA PROGRAMS

1. Write a Java program to print “Hello World”.

AIM

To write a simple Java program that displays the message "**Hello World**" on the screen.

SYNTAX

```
public class ClassName {  
    public static void main(String[] args) {  
        // Statement(s)  
    }  
}
```

PROGRAM

```
public class Practical {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

OUTPUT

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and UnitConverter - Apache NetBeans IDE 25. The central workspace displays the code for 'Practical.java':

```
1  /*  
2  * Click nbfs://nghost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license  
3  * Click nbfs://nghost/SystemFileSystem/Templates/Classes/Class.java to edit this template  
4  */  
5    
6  /**  
7  * @author rohan  
8  */  
9  package practical;  
10   
11  public class Practical {  
12      public static void main(String[] args) {  
13          System.out.println("Hello World");  
14      }  
15  }  
16   
17
```

The Navigator panel on the left shows the 'Members' section with 'Practical' expanded, showing 'Practical()' and 'main(String[]) args'. The Output panel at the bottom shows the build log:

```
> Task :run  
Hello World  
Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.  
You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.  
For more on this, please refer to https://docs.gradle.org/8.10/userguide/command\_line\_interface.html#sec:command\_line\_warnings in the Gradle documentation.  
BUILD SUCCESSFUL in 80ms  
5 actionable tasks: 2 executed, 3 up-to-date
```

At the bottom left, a status bar indicates "Building UnitConverter was successful."

2. Write a Java program to display characters A to Z using a for loop.

AIM

To write a Java program that prints the **uppercase English alphabets (A to Z)** using a for loop.

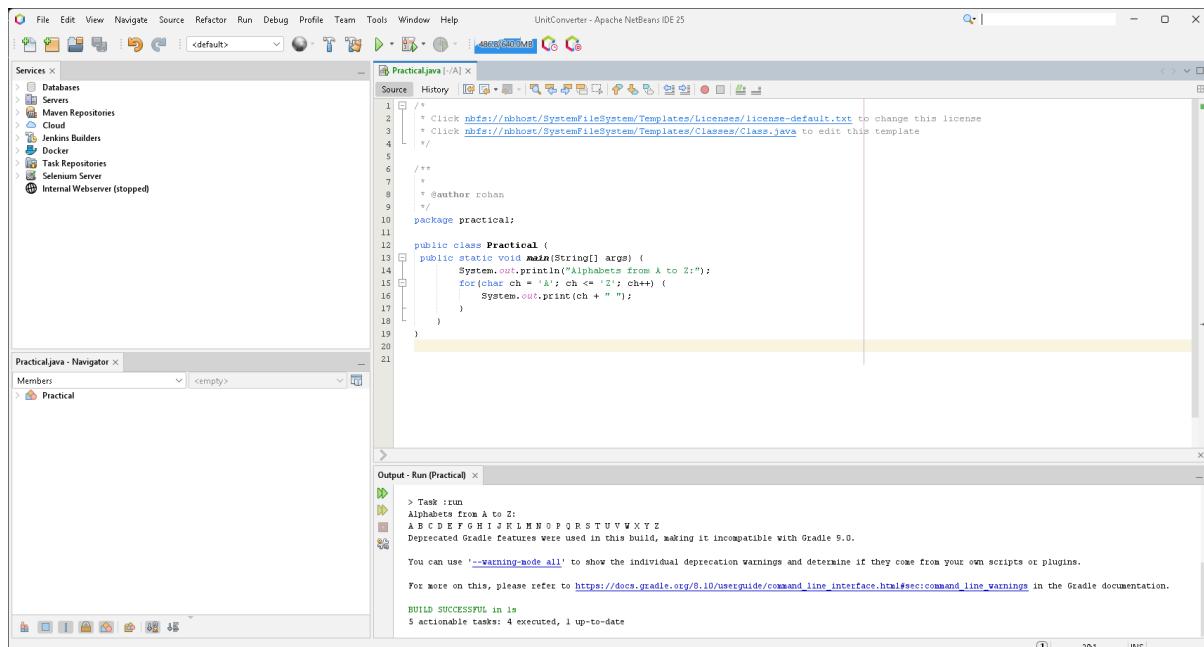
SYNTAX

```
for(char ch = 'A'; ch <= 'Z'; ch++) {  
    // print character  
}
```

PROGRAM

```
public class Practical {  
  
    public static void main(String[] args) {  
  
        System.out.println("Alphabets from A to Z:");  
  
        for(char ch = 'A'; ch <= 'Z'; ch++) {  
  
            System.out.print(ch + " ");  
        }  
    }  
}
```

OUTPUT



The screenshot shows the Apache NetBeans IDE interface. The code editor window displays the following Java code:

```
/*  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template  
 */  
  
package practical;  
  
public class Practical {  
    public static void main(String[] args) {  
        System.out.println("Alphabets from A to Z:");  
        for(char ch = 'A'; ch <= 'Z'; ch++) {  
            System.out.print(ch + " ");  
        }  
    }  
}
```

The Output window below shows the execution results:

```
> Task :run  
Alphabets from A to Z:  
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.  
You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.  
For more on this, please refer to https://docs.gradle.org/8.10/userguide/command\_line\_interface.html#sec:command\_line\_warnings in the Gradle documentation.  
BUILD SUCCESSFUL in 1s  
5 actionable tasks: 4 executed, 1 up-to-date
```

3. Write a Java program to check whether a number is Positive, Negative, or Zero using if...else.

AIM

To write a Java program that determines whether the input number is **positive**, **negative**, or **zero** using **conditional statements**.

SYNTAX

```
if (number > 0)
    // positive
else if (number < 0)
    // negative
else
    // zero
```

PROGRAM

```
import java.util.Scanner;

public class NumberCheck {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        if (num > 0) {
            System.out.println("The number is Positive.");
        } else if (num < 0) {
            System.out.println("The number is Negative.");
        } else {
            System.out.println("The number is Zero.");
        }
    }
}
```

OUTPUT (*Sample Input: 5*)

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and UnitConverter - Apache NetBeans IDE 25. The left sidebar displays services like Databases, Servers, Maven Repositories, Cloud, Jenkins Builders, Docker, Task Repositories, Selenium Server, and Internal Webserver (stopped). The main workspace shows a Java file named NumberCheck.java with the following code:

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5
6  /**
7   *
8   * @author rohan
9   */
10 package practical;
11
12 import java.util.Scanner;
13
14 public class NumberCheck {
15     public static void main(String[] args) {
16         Scanner sc = new Scanner(System.in);
17         System.out.print("Enter a number: ");
18         int num = sc.nextInt();
19
20         if (num > 0) {
21             System.out.println("The number is Positive.");
22         } else if (num < 0) {
23             System.out.println("The number is Negative.");
24         } else {
25             System.out.println("The number is Zero.");
26         }
27     }
28 }
```

The Navigator panel below shows the members of the NumberCheck class. The Output panel at the bottom shows the run results for the NumberCheck task, including the input "Enter a number: 5" and the output "The number is Positive." It also includes a warning about deprecated Gradle features.

4. Write a Java program to find the largest among three numbers using if...else.

AIM

To write a Java program that accepts three numbers as input and determines the **largest** among them using **nested if...else statements**.

SYNTAX

```
if (a > b && a > c)  
    // a is largest  
else if (b > c)  
    // b is largest  
else  
    // c is largest
```

PROGRAM

```
import java.util.Scanner;
```

```
public class LargestNumber {
```

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Enter first number: ");  
    int a = sc.nextInt();
```

```

System.out.print("Enter second number: ");

int b = sc.nextInt();

System.out.print("Enter third number: ");

int c = sc.nextInt();

if (a > b && a > c) {

    System.out.println(a + " is the largest.");

} else if (b > c) {

    System.out.println(b + " is the largest.");

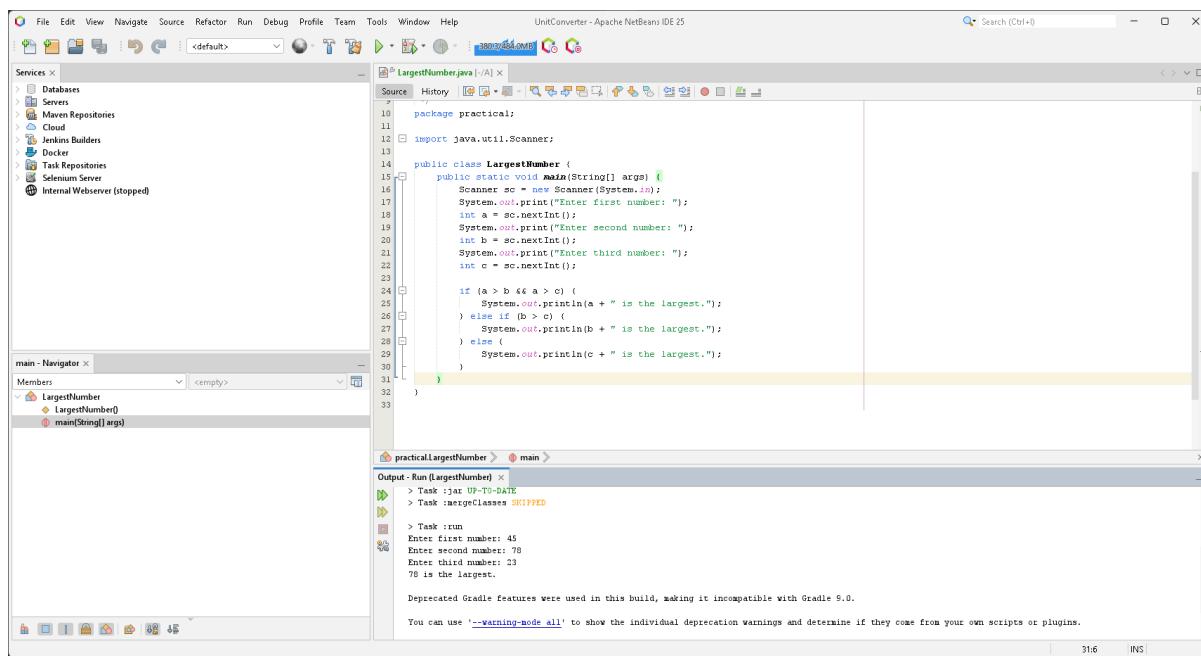
} else {

    System.out.println(c + " is the largest.");

}
}
}
}

```

OUTPUT (Sample Input: 45, 78, 23)



The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a search bar. The left sidebar displays services like Databases, Servers, Maven Repositories, Cloud, Jenkins Builders, Docker, Task Repositories, and Selenium Server. The main workspace shows the code for `LargestNumber.java`. The code uses `Scanner` to read three integers from the user and then prints the largest one. The code editor has syntax highlighting and line numbers. Below the code editor is the Navigator pane, which lists the package `practical`, class `LargestNumber`, and method `main`. The bottom pane shows the Output window, which displays the command-line interface for running the program and its execution results. The results show the program prompting for three numbers (45, 78, 23), then determining that 78 is the largest.

```

package practical;
import java.util.Scanner;
public class LargestNumber {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first number: ");
        int a = sc.nextInt();
        System.out.print("Enter second number: ");
        int b = sc.nextInt();
        System.out.print("Enter third number: ");
        int c = sc.nextInt();

        if (a > b && a > c) {
            System.out.println(a + " is the largest.");
        } else if (b > c) {
            System.out.println(b + " is the largest.");
        } else {
            System.out.println(c + " is the largest.");
        }
    }
}

```

Output - Run (LargestNumber) ×
> Task :jar UP-TO-DATE
> Task :mergeClasses SKIPPED
> Task :run
Enter first number: 45
Enter second number: 78
Enter third number: 23
78 is the largest.

5. Write a Java program to check whether a number is Even or Odd using if...else.

AIM

To write a Java program that determines whether an **input number** is **even** or **odd** using the modulo (%) operator and if...else condition.

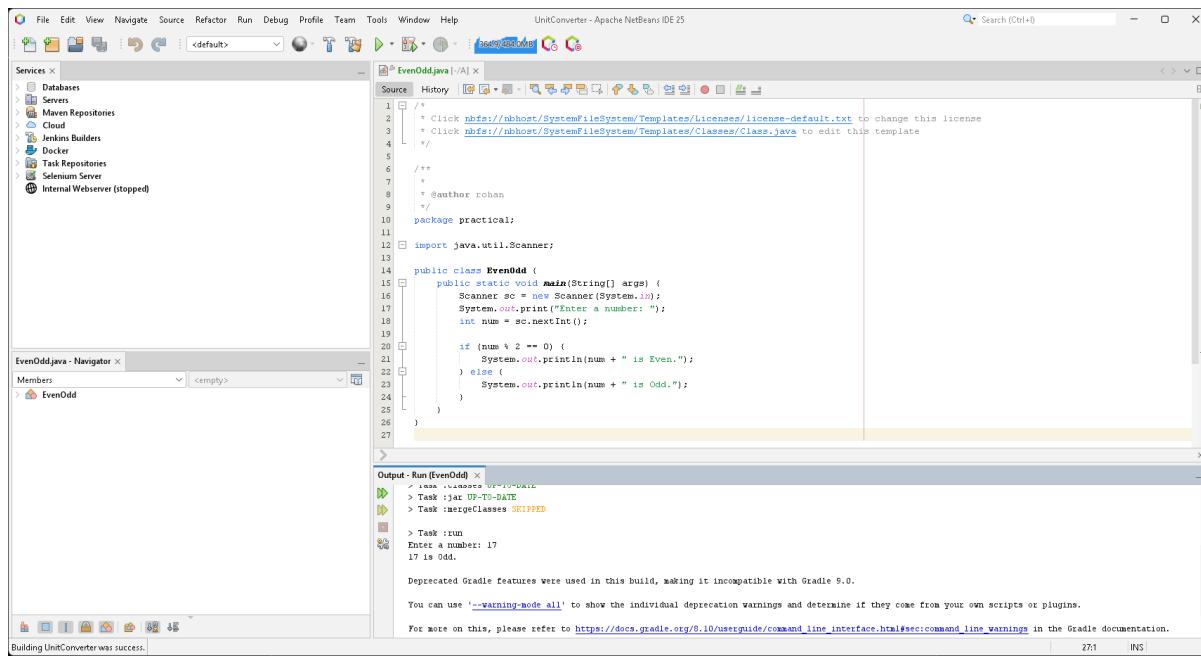
SYNTAX

```
if (number % 2 == 0)  
    // Even  
else  
    // Odd
```

PROGRAM

```
import java.util.Scanner;  
  
public class EvenOdd {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Enter a number: ");  
  
        int num = sc.nextInt();  
  
        if (num % 2 == 0) {  
  
            System.out.println(num + " is Even.");  
  
        } else {  
  
            System.out.println(num + " is Odd.");  
  
        }  
    }  
}
```

OUTPUT (*Sample Input: 17*)



6. Write a Java program to generate a multiplication table using a for loop.

AIM

To write a Java program that **prints the multiplication table** of a given number (from 1 to 10) using a for loop.

SYNTAX

```
for (int i = 1; i <= 10; i++) {  
    System.out.println(number + " * " + i + " = " + (number * i));  
}
```

PROGRAM

```
import java.util.Scanner;  
  
public class MultiplicationTable {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter a number: ");  
        int num = sc.nextInt();  
  
        System.out.println("Multiplication Table of " + num + ":");  
        for (int i = 1; i <= 10; i++) {  
            System.out.println(num + " * " + i + " = " + (num * i));  
        }  
    }  
}
```

OUTPUT (*Sample Input: 5*)

```

MultiplicationTable.java [A]
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5
6  /**
7   * 
8   * @author rohan
9   */
10 package practical;
11
12 import java.util.Scanner;
13
14 public class MultiplicationTable {
15     public static void main(String[] args) {
16         Scanner sc = new Scanner(System.in);
17         System.out.print("Enter a number: ");
18         int num = sc.nextInt();
19
20         System.out.println("Multiplication Table of " + num + ":");
21         for (int i = 1; i <= 10; i++) {
22             System.out.println(num + " * " + i + " = " + (num * i));
23         }
24     }
25 }

```

Output - Run (MultiplicationTable) :

```

> Task :run
Enter a number: 5
Multiplication Table of 5:
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50

```

7. Write a Java program to find the area of a rectangle by taking length and breadth as input.

AIM

To write a Java program that calculates the **area of a rectangle** using the formula $\text{Area} = \text{length} \times \text{breadth}$ based on **user input**.

SYNTAX

`area = length * breadth;`

PROGRAM

```
import java.util.Scanner;
```

```

public class RectangleArea {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter length of the rectangle: ");
        double length = sc.nextDouble();

        System.out.print("Enter breadth of the rectangle: ");
        double breadth = sc.nextDouble();

        double area = length * breadth;
    }
}

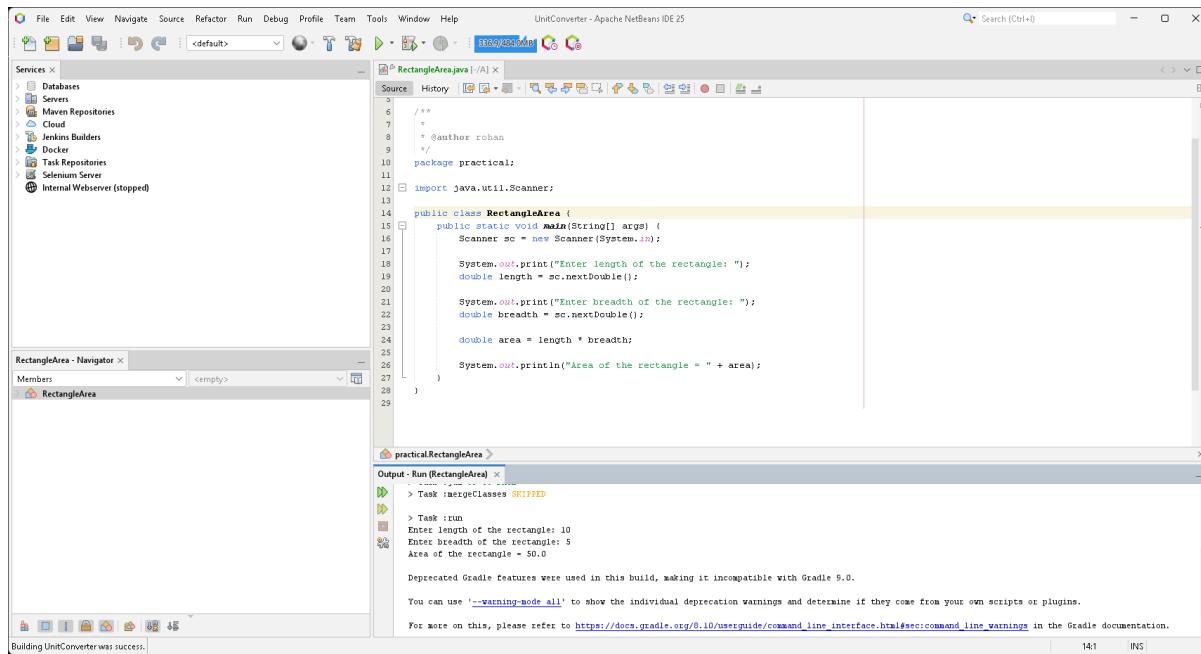
```

```

        System.out.println("Area of the rectangle = " + area);
    }
}

```

OUTPUT (Sample Input: length = 10, breadth = 5)



The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and UnitConverter - Apache NetBeans IDE 25. The main workspace displays a Java file named RectangleArea.java. The code defines a class RectangleArea with a main method that prompts the user for length and breadth, calculates the area, and prints it. The Navigator panel shows the members of the RectangleArea class. The Output panel shows the build and run logs, indicating a successful run with input length=10 and breadth=5, resulting in an area of 50.0.

```

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help UnitConverter - Apache NetBeans IDE 25
Search (Ctrl+F)
Services <default> 336/9480MB
Source History > RectangleArea.java
6 /**
7 * @author rehan
8 */
9 package practical;
10
11 import java.util.Scanner;
12
13 public class RectangleArea {
14     public static void main(String[] args) {
15         Scanner sc = new Scanner(System.in);
16
17         System.out.print("Enter length of the rectangle: ");
18         double length = sc.nextDouble();
19
20         System.out.print("Enter breadth of the rectangle: ");
21         double breadth = sc.nextDouble();
22
23         double area = length * breadth;
24
25         System.out.println("Area of the rectangle = " + area);
26     }
27 }
28
29
practical.RectangleArea >
Output - Run (RectangleArea) >
  > Task :mergeClasses SKIPPED
  > Task run
    > Enter length of the rectangle: 10
    > Enter breadth of the rectangle: 5
    > Area of the rectangle = 50.0
Building UnitConverter was success.

```

8. Write a Java program to print the sum of the first 10 natural numbers.

AIM

To write a Java program that computes and displays the **sum of the first 10 natural numbers** using a for loop.

SYNTAX

```

for (int i = 1; i <= 10; i++) {
    sum += i;
}

```

PROGRAM

```

public class SumNaturalNumbers {
    public static void main(String[] args) {
        int sum = 0;

        for (int i = 1; i <= 10; i++) {

```

```

        sum += i;

    }

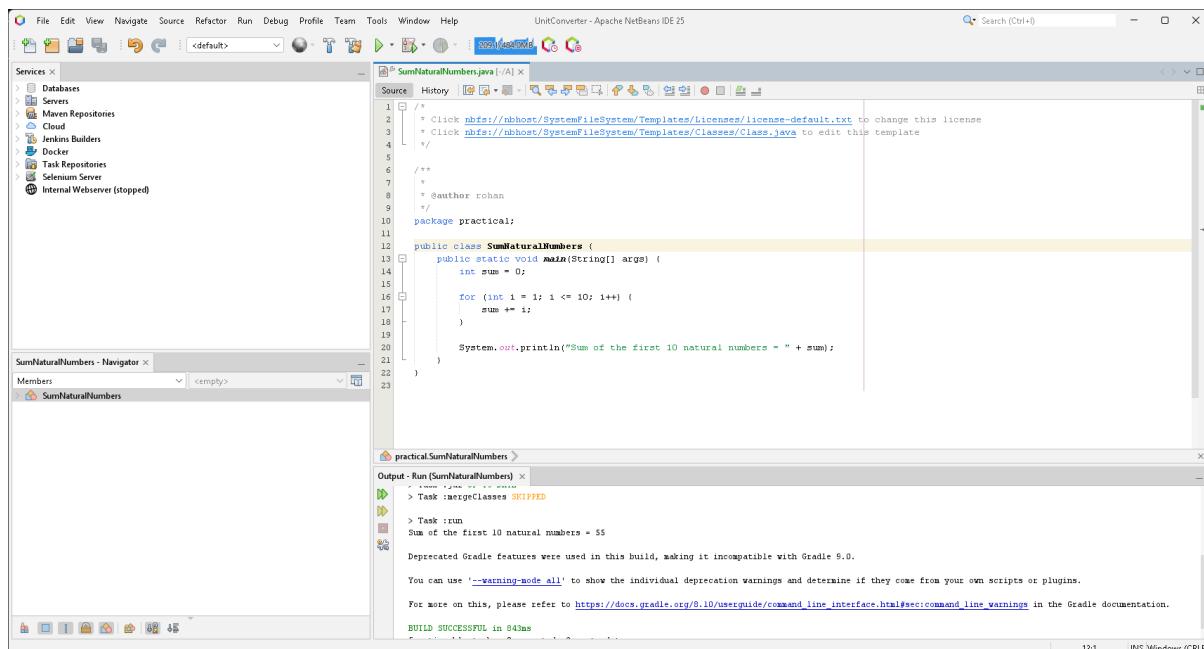
}

System.out.println("Sum of the first 10 natural numbers = " + sum);

}
}

```

OUTPUT



The screenshot shows the Apache NetBeans IDE interface. The main window displays the code for `SumNaturalNumbers.java`. The code calculates the sum of the first 10 natural numbers and prints it to the console. The output window shows the result: `Sum of the first 10 natural numbers = 55`.

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package practical;

public class SumNaturalNumbers {
    public static void main(String[] args) {
        int sum = 0;

        for (int i = 1; i <= 10; i++) {
            sum += i;
        }

        System.out.println("Sum of the first 10 natural numbers = " + sum);
    }
}

```

9. Write a Java program to generate Fibonacci series based on user input.

AIM

To write a Java program that generates the Fibonacci series up to a specified number of terms provided by the user.

SYNTAX

```

int a = 0, b = 1;

for (int i = 1; i <= n; i++) {

    // print a

    int next = a + b;

    a = b;

    b = next;

}

```

PROGRAM

```
import java.util.Scanner;

public class FibonacciSeries {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of terms: ");
        int n = sc.nextInt();

        int a = 0, b = 1;
        System.out.print("Fibonacci Series: ");

        for (int i = 1; i <= n; i++) {
            System.out.print(a + " ");
            int next = a + b;
            a = b;
            b = next;
        }
    }
}
```

OUTPUT (Sample Input: 7)

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and UnitConverter - Apache NetBeans IDE 25. The title bar says "fibonacciSeries [1/A]". The left sidebar has sections for Services (Databases, Servers, Maven Repositories, Cloud, Jenkins Builders, Git, Task Repositories, Selenium Server, Internal Webserver (Stopped)), Navigator (Members: FibonacciSeries), and a search bar. The main workspace contains the code for FibonacciSeries.java. The code is as follows:

```
4  /*
5   */
6  /**
7   * @author rohan
8   */
9  package practical;
10
11 import java.util.Scanner;
12
13 public class FibonacciSeries {
14     public static void main(String[] args) {
15         Scanner sc = new Scanner(System.in);
16         System.out.print("Enter number of terms: ");
17         int n = sc.nextInt();
18
19         int a = 0, b = 1;
20         System.out.print("Fibonacci Series: ");
21
22         for (int i = 1; i <= n; i++) {
23             System.out.print(a + " ");
24             int next = a + b;
25             a = b;
26             b = next;
27         }
28     }
29 }
30
31 practical.FibonacciSeries
```

The Output window shows the run results:

```
> Task :classes UP-TO-DATE
> Task :jdeps UP-TO-DATE
> Task :mergeClasses SKIPPED
> Task :run
Enter number of terms: 7
Fibonacci Series: 0 1 1 2 3 5 8
Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.

For more on this, please refer to https://docs.gradle.org/8.10/userguide/command\_line\_interface.html#sec:command\_line\_warnings in the Gradle documentation.
```

10. Write a Java program to print the factorial of a given number.

AIM

To write a Java program that calculates the **factorial** of a number entered by the user using a for loop.

SYNTAX

```
factorial = 1;  
for (int i = 1; i <= number; i++) {  
    factorial *= i;  
}
```

PROGRAM

```
import java.util.Scanner;  
  
public class Factorial {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter a number: ");  
        int num = sc.nextInt();  
        long factorial = 1;  
  
        for (int i = 1; i <= num; i++) {  
            factorial *= i;  
        }  
  
        System.out.println("Factorial of " + num + " is " + factorial);  
    }  
}
```

OUTPUT (*Sample Input: 5*)

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and UnitConverter - Apache NetBeans IDE 25. The search bar at the top right contains the placeholder "Search (Ctrl+F)".

The left sidebar displays various services: Databases, Servers, Maven Repositories, Cloud, Jenkins Builders, Docker, Task Repositories, Selenium Server, and Internal Webserver (stopped).

The main workspace shows the code editor for "Factorial.java" under the package "practical". The code calculates the factorial of a user input number (5 in this case) and prints the result (120). The code editor has tabs for Source, History, and Diff.

The bottom right pane shows the "Output" window for the "Factorial" task, which includes the command run, standard output (Enter a number: 5, Factorial of 5 is 120), and a deprecation warning about Gradle features.

11. Write a Java program to calculate the sum of digits of a number.

AIM

To find the sum of individual digits of a number using a loop.

SYNTAX

```
while (num > 0) {  
    sum += num %  
    num /= 10;  
}
```

PROGRAM

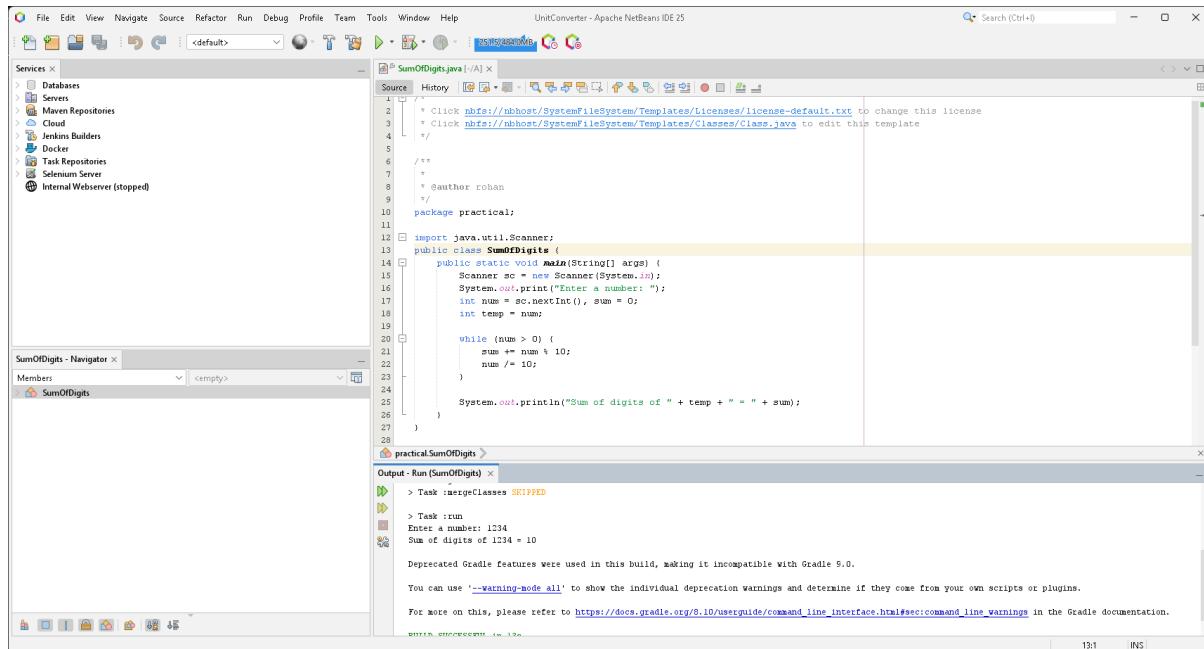
```
import java.util.Scanner;  
  
public class SumOfDigits {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Enter a number: ");  
  
        int num = sc.nextInt(), sum = 0;  
  
        int temp = num;  
  
        while (num > 0) {  
  
            sum += num % 10;  
  
            num /= 10;  
  
        }  
    }  
}
```

```

        System.out.println("Sum of digits of " + temp + " = " + sum);
    }
}

```

OUTPUT



The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar says "UnitConverter - Apache NetBeans IDE 25". The main workspace displays the code for "SumOfDigits.java". The code is as follows:

```

1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Classe.java to edit this template
4  */
5
6 /**
7 * @author rohan
8 */
9
10 package practical;
11
12 import java.util.Scanner;
13 public class SumOfDigits {
14     public static void main(String[] args) {
15         Scanner sc = new Scanner(System.in);
16         System.out.print("Enter a number: ");
17         int num = sc.nextInt(), sum = 0;
18         int temp = num;
19
20         while (num > 0) {
21             sum += num % 10;
22             num /= 10;
23         }
24
25         System.out.println("Sum of digits of " + temp + " = " + sum);
26     }
27 }

```

The "Output" window shows the run results:

```

Output - Run (SumOfDigits) ×
> Task :mergeClasses SKIPPED
> Task :run
Enter a number: 1234
Sum of digits of 1234 = 10

```

Below the code editor, the status bar shows "13:1 INS".

12. Write a Java program to check whether a number is prime.

AIM

To determine whether a number is prime using a loop and conditions.

SYNTAX

```
for (i = 2; i <= n/2; i++) { if (n % i == 0) not prime; }
```

PROGRAM

```

import java.util.Scanner;

public class PrimeCheck {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a number: ");

        int n = sc.nextInt();

        boolean prime = true;

        if (n <= 1) prime = false;

        else {

```

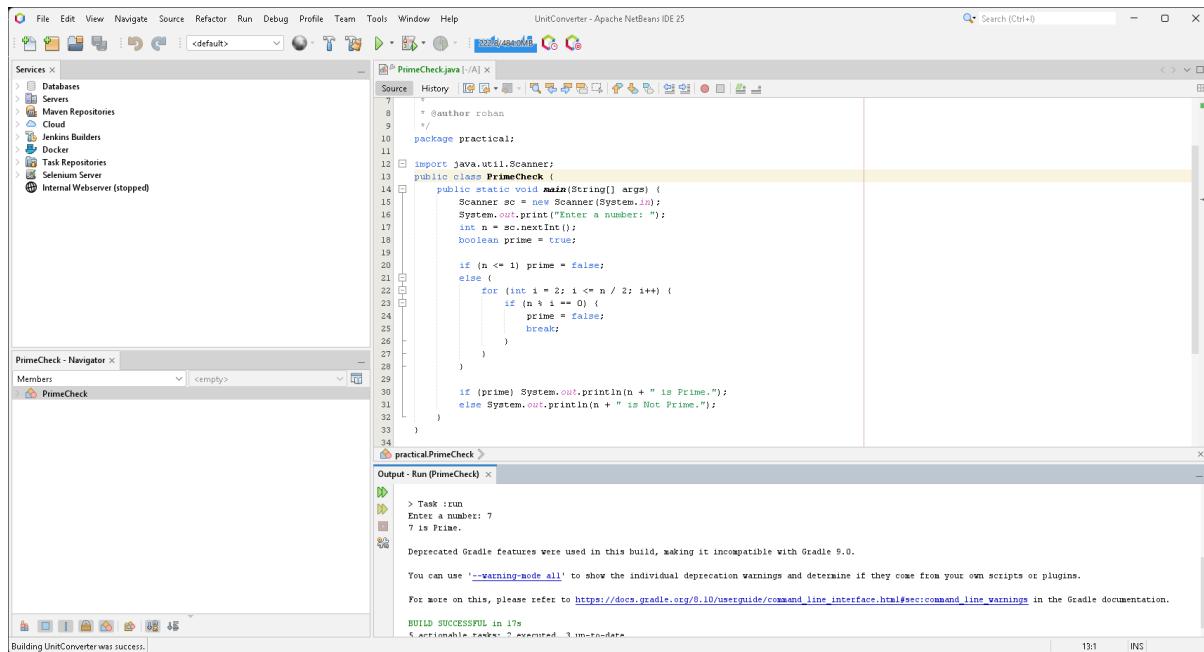
```

for (int i = 2; i <= n / 2; i++) {
    if (n % i == 0) {
        prime = false;
        break;
    }
}

if (prime) System.out.println(n + " is Prime.");
else System.out.println(n + " is Not Prime.");
}

```

OUTPUT



The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and UnitConverter - Apache NetBeans IDE 25. The title bar says "PrimeCheck.java [A] X". The left sidebar has sections for Services (Databases, Servers, Maven Repositories, Cloud, Jenkins Builders, Docker, Task Repositories, Selenium Server, Internal Webserver (stopped)), and a PrimeCheck - Navigator panel showing Members (PrimeCheck). The main workspace displays the Java code for PrimeCheck.java:

```

1  * @author rehan
2  */
3  package practical;
4
5  import java.util.Scanner;
6
7  public class PrimeCheck {
8      public static void main(String[] args) {
9          Scanner sc = new Scanner(System.in);
10         System.out.print("Enter a number: ");
11         int n = sc.nextInt();
12         boolean prime = true;
13
14         if (n <= 1) prime = false;
15         else {
16             for (int i = 2; i <= n / 2; i++) {
17                 if (n % i == 0) {
18                     prime = false;
19                     break;
20                 }
21             }
22         }
23
24         if (prime) System.out.println(n + " is Prime.");
25         else System.out.println(n + " is Not Prime.");
26     }
27 }
28
29
30
31
32
33
34

```

Below the code editor is an Output - Run (PrimeCheck) window showing the terminal output:

```

> Task :run
Enter a number: 7
7 is Prime.

Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.
You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.
For more on this, please refer to https://docs.gradle.org/8.10/userguide/command\_line\_interface.html#sec:command\_line\_warnings in the Gradle documentation.

BUILD SUCCESSFUL in 17s
5 actionable tasks: 5 executed, 0 up-to-date

```

13. Write a Java program to reverse a number.

AIM

To reverse the digits of a number using a loop.

PROGRAM

```

import java.util.Scanner;

public class ReverseNumber {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

```

```

System.out.print("Enter a number: ");

int num = sc.nextInt(), rev = 0;

while (num != 0) {

    rev = rev * 10 + num % 10;

    num /= 10;
}

System.out.println("Reversed number: " + rev);

}

```

OUTPUT

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and UnitConverter - Apache NetBeans IDE 25. The main window displays the code for ReverseNumber.java:

```

3  * Click pbfs://nbhost/SystemFileSystem/Templates/Classes/Classe.java to edit this template
4
5
6 /**
7  * @author rohan
8 */
9
10 package practical;
11
12 import java.util.Scanner;
13 public class ReverseNumber {
14     public static void main(String[] args) {
15         Scanner sc = new Scanner(System.in);
16         System.out.print("Enter a number: ");
17         int num = sc.nextInt(), rev = 0;
18
19         while (num != 0) {
20             rev = rev * 10 + num % 10;
21             num /= 10;
22         }
23
24         System.out.println("Reversed number: " + rev);
25     }
26 }

```

The Navigator panel shows the members of the ReverseNumber class. The Output panel shows the run results:

```

> Task :run
Enter a number: 1234
Reversed number: 4321

Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.
You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.
For more on this, please refer to https://docs.gradle.org/8.10/userguide/command\_line\_interface.html#sec:command\_line\_warnings in the Gradle documentation.

BUILD SUCCESSFUL in 1ms
5 actionable tasks: 2 executed, 3 up-to-date

```

14. Write a Java program to check whether a string is a palindrome.

AIM

To verify if a string is the same forward and backward.

PROGRAM

```

import java.util.Scanner;

public class PalindromeString {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");

```

```

String str = sc.nextLine();

String rev = "";

for (int i = str.length() - 1; i >= 0; i--) {

    rev += str.charAt(i);

}

if (str.equals(rev))

    System.out.println("Palindrome String");

else

    System.out.println("Not a Palindrome");

}
}

```

OUTPUT

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar says "UnitConverter - Apache NetBeans IDE 25". The left sidebar has sections for Services, Maven Repositories, Jenkins Builders, Docker, Task Repositories, Selenium Server, and Internal Webserver (stopped). The main workspace contains three windows: "PalindromeString.java" (Source tab), "PalindromeString - Navigator" (Members tab), and "Output - Run (PalindromeString)". The "PalindromeString.java" window shows Java code for checking if a string is a palindrome. The "Output - Run (PalindromeString)" window shows the console output where the user enters "madam" and the program prints "Palindrome String". A status bar at the bottom right shows "13:1 INS".

15. Write a Java program to count vowels in a string.

AIM

To count the total number of vowels in a given string.

PROGRAM

```

import java.util.Scanner;

public class CountVowels {

    public static void main(String[] args) {

```

```
Scanner sc = new Scanner(System.in);

System.out.print("Enter a string: ");

String str = sc.nextLine().toLowerCase();

int count = 0;

for (int i = 0; i < str.length(); i++) {

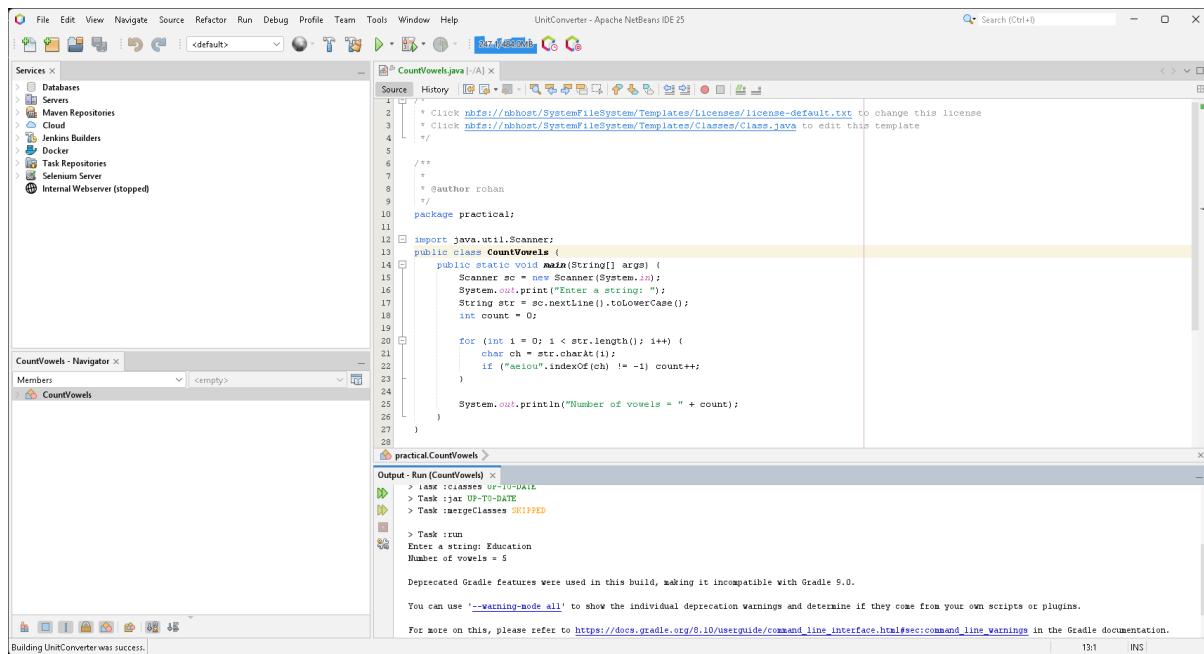
    char ch = str.charAt(i);

    if ("aeiou".indexOf(ch) != -1) count++;

}

System.out.println("Number of vowels = " + count);
```

OUTPUT



OPERATING WEB BASED APPLICATION

Case Study 1: Online Bill Calculator / Book Rail Ticket

AIM

To explore and demonstrate the use of **web-based utility applications** such as **Online Electricity Bill Calculators** or **Online Railway Ticket Booking Systems**, enabling users to calculate fares, charges, or book services online.

DESCRIPTION

Web-based billing and booking systems are used by government and private service providers. These platforms allow users to:

- Input parameters such as units consumed, class of travel, distance, etc.
- Automatically calculate charges/tickets in real-time.
- Receive confirmation through web forms, email/SMS, or e-payment integration.

TOOLS USED

- **Web Browser** (Chrome / Firefox)
- **Official Web Portals** (IRCTC, State Electricity Boards, etc.)
- **Internet Connectivity**

OUTPUT

The screenshot shows the IRCTC website interface. At the top, there are navigation links for REFUND STATUS, LOGIN, CONTACT US, HELP & SUPPORT, DAILY DEALS, ALERTS, and a date range (07-Jul-2025 [B2255]). There are also icons for IRCTC EXCLUSIVE, TRAINS, LOYALTY, IRCTC PAYMENT, BUSSES, FLIGHTS, HOTELS, HOLIDAYS, MEALS, PROMOTIONS, MY ACCOUNT, and MORE. A banner at the bottom of the page reads "TRAINS AT A GLANCE-2025 With Effect From 1 January 2025 ₹100". The main content area includes sections for "BOOK TICKET", "Last Transaction Detail" (No last transaction), and "Upcoming Journey" (No upcoming journeys). Below these sections is a large image of a white and blue high-speed train. At the bottom, there is a promotional banner for IRCTC Pay and a section titled "Have you not found the right one? Find a service suitable for you here." with icons for FLIGHTS, HOTELS, RAIL DRISHTI, E-CATERING, and BUS. A small footer at the very bottom right includes links for "DEALS OF THE DAY", "Get Train Ticket", and "Get Train Ticket App".

Conclusion

Web-based utilities simplify real-world services, save time, ensure accuracy, and reduce human error.

Case Study 2: Online Quiz – Internet and Phone Safety for School Children

AIM

To use an **online quiz platform** to educate school children about **internet safety and phone security**, and to test their awareness.

DESCRIPTION

- The quiz includes topics like:
 - Strong password creation
 - Avoiding suspicious links
 - Not sharing OTPs
 - Cyberbullying awareness
 - Students access the quiz via platforms such as **Google Forms**, **Kahoot**, or **School LMS portals**.
-

TOOLS USED

- Online Quiz Tools (Google Forms, Kahoot, etc.)
 - Web Browser
 - Internet Access
-

OUTPUT

The screenshot shows a course assessment interface. At the top, there's a header with the Reliance Foundation Skilling Academy logo and navigation links for Skill Development, Job Opportunities, Mentorship, About Us, and user profile. The main area is titled "Course Assessment" under "Skill Development / IoT Network Specialist Certificate Programme / Course Assessment". A question Q1 asks: "What is one of the primary goals of network assessment and optimisation?" with four options: "Reducing power consumption", "Increasing latency", "Blocking all devices", and "Ensuring the network operates at its optimal capacity". The fourth option is checked and highlighted in green. Below the question is a "Submit" button. To the right, a sidebar titled "Content" lists three modules: "1.IoT Network Design" (Completed | 0%), "2.IoT Network Management and Optimization" (Completed | 0%), and "3.IoT Projects Documentation" (Completed | 0%). At the bottom, there are "Previous Lesson" and "Final" buttons, along with navigation icons for back, forward, and search.

Conclusion

The quiz helps improve awareness about cyber safety and responsible internet usage among students.

Case Study 3: Online Game – Hangman

AIM

To use a **web-based game** to understand the logic and programming principles behind classic word games like **Hangman**.

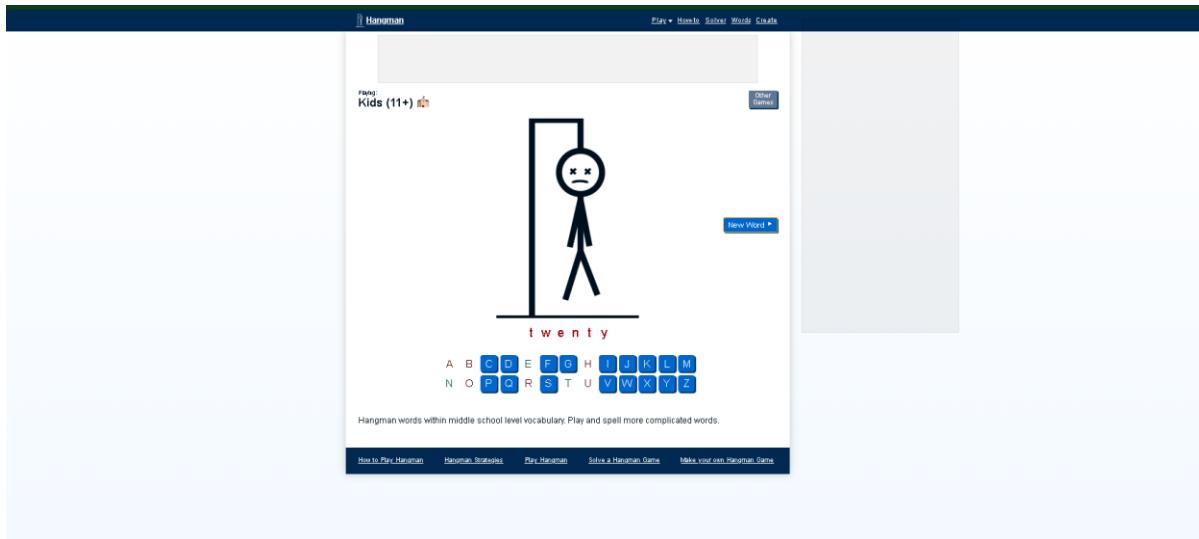
DESCRIPTION

- The game involves guessing a word one letter at a time.
 - Each incorrect guess reveals a part of the hangman graphic.
 - Educational value: vocabulary building, logical reasoning.
-

TOOLS USED

- Online platforms like <https://www.hangmanwords.com/>
 - JavaScript or Python (for DIY versions)
 - Web Browser
-

OUTPUT



Conclusion

Online games like Hangman are not only entertaining but also reinforce logical and language skills using simple web technologies.

BIBLIOGRAPHY

The following sources were referred to during the preparation of this project file:

1. Textbooks:

- NCERT Computer Science Class XII
- CBSE Informatics Practices Curriculum Guide (2025–26)

2. Online Resources:

- <https://www.w3schools.com/sql/>
- <https://www.javatpoint.com/java-programs>
- <https://docs.oracle.com/javase/>
- <https://www.mysql.com/>
- <https://www.geeksforgeeks.org/>
- <https://www.chatgpt.com/>

3. Software & Tools:

- MySQL Server & Workbench
- Java JDK (version 17 or higher)
- Visual Studio Code / Notepad++
- Web Browsers (Chrome, Firefox)

4. Teacher & Peer Support:

- Classroom Lectures
- Practical Lab Sessions
- Guidance from subject teacher **Mr. Raju Gupta**



APPENDIX / ANNEXURE

Supporting content and additional information attached with this project:

1. Screenshots:

- MySQL Command Prompt Outputs
- Java Program Output Snapshots
- Web Application Interfaces (Online Quiz, Bill Calculator, Hangman Game)

2. Source Code:

- Complete Java source code (.java files)
- SQL script files used in practicals

3. Test Data & Results:

- Sample input/output used for program validation
- Data sets used in SQL queries (e.g., student table entries)

4. Certificates & Documentation:

- Project Certificate Page
 - Acknowledgement Page
 - CBSE Practical Guidelines (if attached) [802-IT.pdf](#)
-