1. What are the four pillars of OOP?

The four pillars are:

Encapsulation: Bundling data (attributes) and methods that operate on that data into a single unit or class.

Abstraction: Hiding complex implementation details and showing only the necessary features.

Inheritance: Allowing a new class to inherit properties and behaviors (methods) from an existing class.

Polymorphism: Allowing objects of different classes to be treated as objects of a common base class, mainly via method overriding (runtime polymorphism) and method overloading (compile-time polymorphism).

2. What is a class and an object?

A **class** is a blueprint for creating objects. It defines a data structure and behavior (methods).

An **object** is an instance of a class. It has state (attributes) and behavior (methods) defined by the class.

3. What is the difference between method overloading and method overriding?

Method overloading: Defining multiple methods in the same class with the same name but different parameter types or numbers (compile-time polymorphism).

Method overriding: When a subclass provides a specific implementation of a method that is already defined in its superclass (runtime polymorphism).

4 .What are some advantages of using OOPs?

- ❖ OOPs is very helpful in solving very complex level of problems.
- ❖ Highly complex programs can be created, handled, and maintained easily using object-oriented programming.
- ❖ OOPs, promote code reuse, thereby reducing redundancy.
- ❖ OOPs also helps to hide the unnecessary details with the help of Data Abstraction.

- OOPs, are based on a bottom-up approach, unlike the Structural programming paradigm, which uses a top-down approach.
- Polymorphism offers a lot of flexibility in OOPs.

7. What are access specifiers and what is their significance?

Access specifiers, as the name suggests, are a special type of keywords, which are used to control or specify the accessibility of entities like classes, methods, etc. Some of the access specifiers or access modifiers include “private”, “public”, etc. These access specifiers also play a very vital role in achieving Encapsulation - one of the major features of OOPs

8 . What is __init__?

__init__ is a contructor method in Python and is automatically called to allocate memory when a new object/instance is created. All classes have a __init__ method associated with them. It helps in distinguishing methods and attributes of a class from local variables.

9 . What are the various types of inheritance?

The various types of inheritance include:

- Single inheritance
- Multiple inheritances
- Multi-level inheritance
- Hierarchical inheritance
- Hybrid inheritance

10 . What is a subclass?

The subclass is a part of Inheritance. The subclass is an entity, which inherits from another class. It is also known as the child class.

11 .What is encapsulation? Why is it important?

Encapsulation refers to restricting direct access to an object's data by making attributes private and providing public getter/setter methods. It ensures data integrity and security, as you control how the data is accessed and modified.

12 . Explain the concept of inheritance.

Inheritance allows one class (child or derived class) to inherit properties and methods from another class (parent or base class), enabling code reusability and hierarchical relationships between classes.

13 . What is polymorphism in OOP?

Polymorphism means "many forms" and allows objects of different classes to be treated as objects of a common base class. It mainly occurs through:

- **Method Overriding**: Providing different implementations of a method in a subclass.
- **Method Overloading**: Multiple methods with the same name but different signatures.

14 . What is a constructor? Can you explain its purpose?

A constructor is a special method that is automatically invoked when an object of a class is created. Its purpose is to initialize the object's state by setting initial values for its attributes.

15 . What is an abstract method?

An abstract method is a method that is declared but does not have an implementation in the base class. It is meant to be overridden by subclasses.

16 . What is a destructor in OOP?

A destructor is a method that is called when an object is destroyed or goes out of scope. It is used to release resources such as memory or file handles that were allocated by the object.

17 . A public method example show me :

```
class Person:
    def __init__(self, name) -> None:
        self.name = name

p1 = Person("Merazul Islam")

print(p1.name)

p1.name = "meraz"

print(p1.name)
```

18 . What is SQL?

SQL (Structured Query Language) is a standard programming language used for managing and manipulating relational databases. It is used for querying, inserting, updating, and deleting data, as well as creating and modifying the structure of database systems.

19 .What is a primary key?

A primary key is a unique identifier for a record in a table. It ensures that no two rows have the same value in the primary key column(s). A primary key cannot contain `NULL` values and must have a unique value for each row.

20. What is a foreign key?

A foreign key is a field (or collection of fields) in one table that refers to the primary key in another table. It is used to establish a link between the data in two tables and enforce referential integrity.

21 . What is a database?

A database is a systematically organized collection of data arranged into tables composed of rows and columns. The primary purpose of databases is to efficiently store, manage, and retrieve data.

22 . What do you mean by a NULL value in SQL?

A NULL value in SQL represents the absence of data in a column. It is not the same as an empty string or zero; it signifies that the data is missing or unknown. NULL values can be used in columns with optional data or when the actual data is unavailable.

23.What is a table and a field in SQL?

In SQL, a table is a structured data collection organized into rows and columns. Each column in a table is called a field, representing a specific attribute or property of the data.

24. What is the SELECT statement?

The SELECT command gets zero or more rows from one or more database tables or views. SELECT the most frequent data manipulation language (DML) command in most applications. SELECT queries define a result set, but not how to calculate it, because SQL is a declarative programming language.

25 .. What are the different types of joins in SQL?

- **INNER JOIN**: Returns rows that have matching values in both tables.
- **LEFT (OUTER) JOIN**: Returns all rows from the left table and the matching rows from the right table. If there is no match, NULL values are returned for columns from the right table.
- **RIGHT (OUTER) JOIN**: Returns all rows from the right table and the matching rows from the left table. If there is no match, NULL values are returned for columns from the left table.

- **FULL (OUTER) JOIN**: Returns rows when there is a match in either table. If there is no match, NULL values are returned for unmatched rows.

26 .How do you update a value in SQL?

The UPDATE statement serves the purpose of altering pre-existing records within a table. It involves specifying the target table for the update, the specific columns to be modified, and the desired new values to be applied. For example:

```
UPDATE employees SET salary = 60000 WHERE department = 'IT';
```

27. . What is a subquery? Provide an example.?

A subquery refers to a query that is embedded within another query, serving the purpose of fetching information that will subsequently be employed as a condition or value within the encompassing outer query. For example, to find employees with salaries greater than the average salary:

```
SELECT name

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);
```

28 .What is the difference between DELETE and TRUNCATE?

DELETE: Removes specific rows from a table based on a condition. It can be rolled back and triggers are activated. It is slower since it logs individual row deletions.

TRUNCATE: Removes all rows from a table without logging individual row deletions. It cannot be rolled back (in some systems) and is faster but less flexible, as it doesn't trigger `ON DELETE` actions.

29.What is denormalization?

Denormalization is the process of combining tables or introducing redundancy in a database to improve read performance, often used in data warehousing or analytical systems.

30 .What are aggregate functions?

COUNT(): Returns the number of rows.

SUM(): Returns the total sum of a numeric column.

AVG(): Returns the average value of a numeric column.

MAX(): Returns the maximum value.

MIN(): Returns the minimum value.

31 .What are triggers in SQL?

A trigger is a stored procedure that automatically executes or fires when a specific event occurs in the database, such as `INSERT`, `UPDATE`, or `DELETE`. Triggers can be used for maintaining data integrity or logging changes.

32. What is the difference between `DROP`, `DELETE`, and `TRUNCATE`?

- **DELETE**: Removes rows based on a condition. It can be rolled back and triggers are fired.
- **TRUNCATE**: Removes all rows from a table without logging individual row deletions. It cannot be rolled back (in some systems).
- **DROP**: Deletes the table or database entirely, including its structure and data.