**23. Difference between inheritance and encapsulation.**

**Inheritance:**

* **Definition:** Inheritance is a mechanism that allows a class to inherit properties and behaviours from another class.
* **Purpose:** It promotes code reusability by allowing a new class to use the existing functionality of another class. The new class can also extend or override the inherited methods to provide specialized behaviour.
* **Key Concepts:** Base Class (Superclass), Derived Class (Subclass).

**Encapsulation:**

* **Definition:** Encapsulation is the bundling of data and methods that operate on the data into a single unit, known as a class. It restricts direct access to some of an object's components and can prevent the accidental modification of data.
* **Purpose:** It helps in hiding the internal details of an object and only exposes what is necessary. This aids in creating more robust and maintainable code by providing a clear interface for interacting with an object.
* **Key Concepts:** Public, Private, and Protected Members.

**Q. Difference between inheritance and abstraction.**

**Inheritance:**

* **Definition:** Inheritance is a mechanism that allows a class to inherit properties and behaviours from another class.
* **Purpose:** It promotes code reusability by allowing a new class to use the existing functionality of another class. The new class can also extend or override the inherited methods to provide specialized behaviour.
* **Key Concepts:** Base Class (Superclass), Derived Class (Subclass).

**Abstraction:**

* **Definition**: Abstraction is used for hiding the unwanted data and giving relevant data.
* **Purpose:** Abstraction lets you focus on what the object does instead of how it does it.
* **Key Concepts:** Abstract Classes and Methods, Interfaces.