**8. Explain the difference between HashSet and HashMap?**

In Java, both `HashSet` and `HashMap` are important data structures used for efficient data storage and retrieval.

`HashSet` is utilized when you need to store a collection of unique values, t. For instance, storing unique employee numbers or any other identifiers is a common use case for `HashSet`. It's a class that implements the Set interface, ensuring that it only contains unique elements. `HashSet` actually uses a `HashMap` to store its objects. The elements are stored as keys in the `HashMap`, and the value associated with each key is a constant dummy value. The use of `HashSet` ensures that duplicates are automatically eliminated.

On the other hand, `HashMap` is employed when you need to store a set of unique key-value pairs, creating a two-dimensional relationship. For example, associating employee numbers with corresponding employee names is a suitable use case for `HashMap`. It implements the Map interface, allowing you to store and retrieve values based on their associated keys. In the context of relational objects, such as incident numbers mapped to their respective statuses.

The fundamental mechanism behind both `HashSet` and `HashMap` involves the use of a hashing algorithm. Hashing allows for quick access to elements based on their hashed keys. In `HashSet`, the elements are hashed and stored in the `HashMap` as keys with dummy values, while in `HashMap`, both keys and their associated values are hashed for efficient retrieval. This hashing mechanism significantly enhances the speed of data retrieval. Understanding when to use `HashSet` for unique values and `HashMap` for key-value pair associations is fundamental in optimizing data management and retrieval in Java applications.