	PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE -43		
	Department of Electronics and Telecommunication Engineering		
	ASSESSMENT YEAR: - 2021-22		**CLASS: - TE-V**
	Subject: - Advanced Java Programming		
Expt. No: 13	**LAB Ref: ETC/2021-22/**	**ROLL NO:32147**	**SUBMISION DATE:**

Title: -

Problem Statement: -

Write a code in java for creating a HashMap for student data which has Student roll number as key and Name as value and print all the students details.

Objectives: -

- Create a HashMap to store data of students
- Which will have Roll no. as a key

Theory (Write Theory of the new concept demonstrated in this Assignment)

HashMap<K, V> is a part of Java's collection since Java 1.2. This class is found in **java.util** package. It provides the basic implementation of the Map interface of Java. It stores the data in **(Key, Value)** pairs, and you can access them by an index of another type (e.g. an Integer). One object is used as a key (index) to another object (value). If you try to insert the duplicate key, it will replace the element of the corresponding Key.

HashMap is similar to HashTable, but it is unsynchronized. It allows to store the null keys as well, but there should be only one null key object and there can be any number of null values. This class makes no guarantees as to the order of the map. To use this class and its methods, you need to import java.util.HashMap package or its superclass.

Constructors in HashMap is as follows:
HashMap provides 4 constructors and the access modifier of each is public which are listed as follows:

1. HashMap()
2. HashMap(int initialCapacity)
3. HashMap(int initialCapacity, float loadFactor)
4. HashMap(Map map)

Performing Various Operations on HashMap

1. Adding Elements: In order to add an element to the map, we can use the put() method. However, the insertion order is not retained in the Hashmap. Internally, for every element, a separate hash is generated and the elements are indexed based on this hash to make it more efficient.

2. Changing Elements: After adding the elements if we wish to change the element, it can be done by again adding the element with the put() method. Since the elements in the map are indexed using the keys, the value of the key can be changed by simply inserting the updated value for the key for which we wish to Change.

3. Removing Element: In order to remove an element from the Map, we can use the remove() method. This method takes the key value and removes the mapping for a key from this map if it is present inthe map

4. Traversal of HashMap: We can use the Iterator interface to traverse over any structure of the Collection Framework. Since Iterators work with one type of data we use Entry< ? , ? > to resolve the two separate types into a compatible format. Then using the next() method we print the entries of HashMap

Diagram: -

Learning Outcomes: -

	1	Learned how to create HashMap.
	2	Learned how to insert values in HashMap.
	3	Learned how to iterate HashMap.
	4	Learned how to modify HashMap.
	5	

Continuous Assessment			
RPP (out of 5)	SPO (out of 5)	Total (Out of 10)	Sign
			Date: -

#(RPP – Regularity, Punctuality, Performance), (SPO – Submission, Presentation, Oral)

Important Questions: -
1. What is java collection?
2. Explain HashMap in detail.
3. Explain different constructors which is provided by HashMap.
4. What are the methods in HashMap?
5. Differentiate between HashMap and HashTable.