MET Bhujbal Knowledge City
MET INSTITUTE OF COMPUTER SCIENCE

Advanced Data Structures Lab
ACADEMIC YEAR 2025-2026

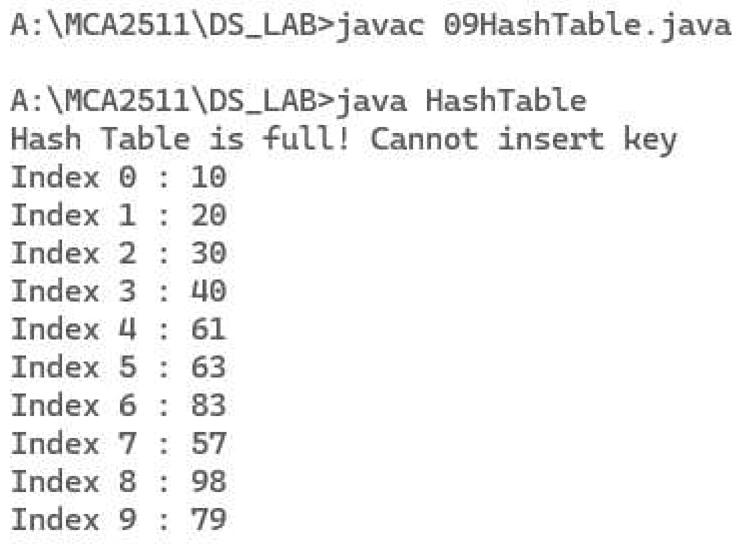
MCA Semester 1	Subject : Advanced Data Structures Lab
Name : Mukund Gangurde	Topic: HashingTech
Roll No. : MCA2511	Date : 30-10-2025

1) Hashing with Modulo Division & Linear Probe

Code:
09HashTable.java
```
class HashTable
{
	private Integer[] table;
	private int size;
	private int capacity;

	public HashTable(int capacity)
	{
		this.capacity = capacity;
		size = 0;
		table = new Integer[capacity];
	}

	//Hash Function - Modulo Division
	private int hash(int key)
	{
		return key%capacity;
	}//end of hash

	//Insert key using Hash Function
	public void insert(int key)
	{
		if(size >= capacity)
		{
			System.out.println("Hash Table is full! Cannot insert key");
			return;
		}

		int index = hash(key);
		while(table[index] != null)
		{
			//Linear Passing
			index = (index+1) % capacity;
		}
```

```
        table[index] = key;
        size++;
    }//end of insert

    //Display the Hash Table
    public void display()
    {
        for(int i=0; i<capacity; i++)
        {
            if(table[i] != null)
            {
                System.out.println("Index " + i + " : " + table[i]);
            }
            else
            {
                System.out.println("Index " + i + " : null");
            }
        }//end of for
    }//end of display

    public static void main(String[] args)
    {
        HashTable h = new HashTable(10);

        //Sample keys to insert
        int[] keys = {10,20,30,40,57,61,63,79,83,98,99};

        for(int key:keys)
        {
            h.insert(key);
        }

        //Display
        h.display();
    }//end of psvm
}//end of HashTable
```

Output:



```
A:\MCA2511\DS_LAB>javac 09HashTable.java

A:\MCA2511\DS_LAB>java HashTable
Hash Table is full! Cannot insert key
Index 0 : 10
Index 1 : 20
Index 2 : 30
Index 3 : 40
Index 4 : 61
Index 5 : 63
Index 6 : 83
Index 7 : 57
Index 8 : 98
Index 9 : 79
```

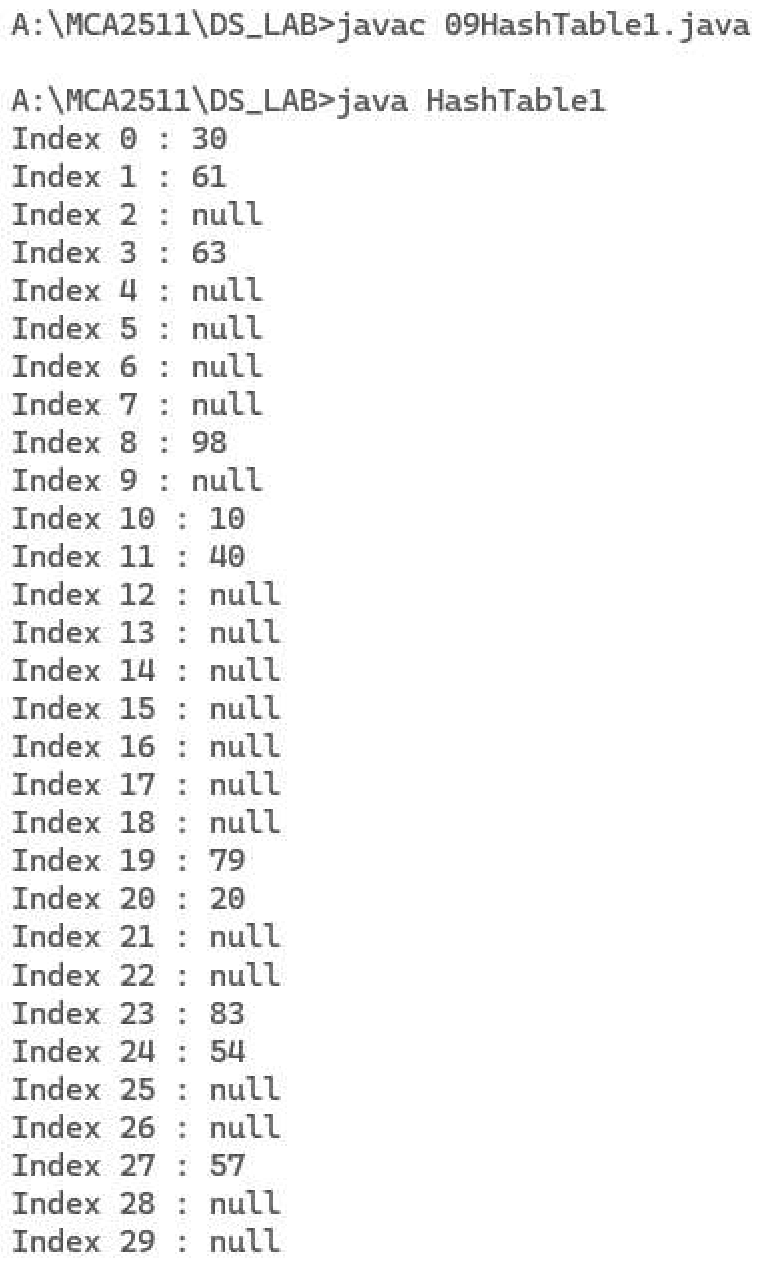
2) Hashing with Digit Extraction & Linear Probe

Code:

<u>09HashTable1.java</u>

```
class HashTable1
{
	private Integer[] table;
	private int size;
	private int capacity;

	public HashTable1(int capacity)
	{
		this.capacity = capacity;
		size = 0;
		table = new Integer[capacity];
	}

	//Hash Function - Modulo Division
	private int hash(int key)
	{
		return (key%100)%capacity;
	}//end of hash

	//Insert key using Hash Function
	public void insert(int key)
	{
		if(size >= capacity)
		{
			System.out.println("Hash Table is full! Cannot insert key");
			return;
		}

		int index = hash(key);
		while(table[index] != null)
		{
			//Linear Passing
			index = (index+1) % capacity;
		}

		table[index] = key;
		size++;
	}//end of insert

	//Display the Hash Table
	public void display()
	{
		for(int i=0; i<capacity; i++)
		{
			if(table[i] != null)
			{
				System.out.println("Index " + i + " : " + table[i]);
			}
			else
			{
				System.out.println("Index " + i + " : null");
			}
		}//end of for
	}//end of display

	public static void main(String[] args)
	{
		HashTable1 h = new HashTable1(20);

		//Sample keys to insert
		int[] keys = {10,20,30,40,57,61,63,79,83,98,54};

		for(int key:keys)
		{
			h.insert(key);
		}

		//Display
		h.display();
	}//end of psvm
}//end of HashTable
```

Output:



```
A:\MCA2511\DS_LAB>javac 09HashTable1.java

A:\MCA2511\DS_LAB>java HashTable1
Index 0 : 30
Index 1 : 61
Index 2 : null
Index 3 : 63
Index 4 : null
Index 5 : null
Index 6 : null
Index 7 : null
Index 8 : 98
Index 9 : null
Index 10 : 10
Index 11 : 40
Index 12 : null
Index 13 : null
Index 14 : null
Index 15 : null
Index 16 : null
Index 17 : null
Index 18 : null
Index 19 : 79
Index 20 : 20
Index 21 : null
Index 22 : null
Index 23 : 83
Index 24 : 54
Index 25 : null
Index 26 : null
Index 27 : 57
Index 28 : null
Index 29 : null
```