

```

/Program on HashMap<K,V>
package com.ravi.hash_map;

import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;

public class HashMapDemo1
{
    public static void main(String[] args)
    {
        HashMap<Integer, String> map = new HashMap<>();

        map.put(1, "Vanilla");
        map.put(2, "Butterscotch");
        map.put(3, "Chocolate");
        map.put(4, "Cotton Candy");

        System.out.println("By using toString() Method :");
        System.out.println("HashMap: " + map); //{key = value}

        System.out.println("By using forEach() Method :");
        map.forEach((key, value)-> System.out.println(key + " -> "+value));

        String value = map.get(2);
        System.out.println("Value for key 2: " + value);

        value = map.getDefault(3, "Key is not available");
        System.out.println("Value for key 3: " + value);

        boolean hasKey = map.containsKey(3);
        System.out.println("HashMap contains key 3: " + hasKey);

        boolean hasValue = map.containsValue("Vanilla");
        System.out.println("HashMap contains value 'Vanilla': " + hasValue);

        map.remove(1);
        System.out.println("HashMap after removing key 1: " + map);

        System.out.println("Iterating through HashMap:");

        for(Map.Entry<Integer, String> entry : map.entrySet())
        {
            System.out.println(entry.getKey()+" -> "+entry.getValue());
        }

        System.out.println("Iterating through Iterator");
        Iterator<Entry<Integer, String>> itr = map.entrySet().iterator();
        itr.forEachRemaining(System.out::println);

        int size = map.size();
        System.out.println("Size of HashMap: " + size);

        map.clear();
        System.out.println("HashMap after clearing: " + map); //{ }
        System.out.println("Is Map empty :"+map.isEmpty());
    }
}

package com.ravi.hash_map;

import java.util.HashMap;

public class HashMapDemo2
{
    public static void main(String[] args)
    {
        HashMap<Integer, String> studentRecords = new HashMap<>();

        studentRecords.put(101, "Scott");
        studentRecords.put(102, "Smith");
        studentRecords.put(103, "Martin");
        studentRecords.put(104, "Aryan");

        System.out.println("Student Records: " + studentRecords);

        int searchId = 103;
        String studentName = studentRecords.get(searchId);

        if (studentName != null)
        {
            System.out.println("Student with ID " + searchId + " is " + studentName);
        }
        else
        {
            System.out.println("Student with ID " + searchId + " not found.");
        }

        System.out.println(studentRecords.put(103, "Rahul"));
        System.out.println("Updated Records: " + studentRecords);

        studentRecords.remove(104);
        System.out.println("Records after removal: " + studentRecords);

        int idToCheck = 101;
        System.out.println("Does ID " + idToCheck + " exist? " +
studentRecords.containsKey(idToCheck));

        String nameToCheck = "Aryan";
        System.out.println("Does name '" + nameToCheck + "' exist? " +
studentRecords.containsValue(nameToCheck));

        System.out.println("Iterating through records:");
        for(HashMap.Entry<Integer, String> entry : studentRecords.entrySet())
        {
            System.out.println("ID: " + entry.getKey() + ", Name: " + entry.getValue());
        }

        studentRecords.clear();
        System.out.println("All records cleared: " + studentRecords); //{ }
    }
}

package com.ravi.hash_map;

import java.util.Collection;
import java.util.HashMap;
import java.util.Set;

public class HashMapDemo3
{
    public static void main(String[] args)
    {
        HashMap<Integer,String> newmap1 = new HashMap<>();

        HashMap<Integer,String> newmap2 = new HashMap<>();

        newmap1.put(1, "OCPJP");
        newmap1.put(2, "is");
        newmap1.put(3, "best");

        System.out.println("Values in newmap1: " + newmap1);

        newmap2.put(4, "Exam");

        newmap2.putAll(newmap1);

        System.out.println("Iterating through forEach()");
        newmap2.forEach((k,v)->System.out.println(k+ " : "+v));

        System.out.println("All the Unique keys");
        Set<Integer> setOfKeys = newmap2.keySet();
        System.out.println(setOfKeys);

        System.out.println("All the values");
        Collection<String> values = newmap2.values();
        System.out.println(values);

        System.out.println(".....");

        System.out.println("Loose Coupling for Merging one Map to another");

        HashMap<Integer, String> hm1 = new HashMap<>();

        hm1.put(1, "Ravi");
        hm1.put(2, "Rahul");
        hm1.put(3, "Rajen");

        HashMap<Integer, String> hm2 = new HashMap<>(hm1);

        System.out.println("Mapping of HashMap hm1 are : " + hm1);

        System.out.println("Mapping of HashMap hm2 are : " + hm2);

    }
}

package com.ravi.hash_map;

import java.util.HashMap;

record Employee(Integer empId, String empName)
{
}

public class HashMapDemo4
{
    public static void main(String[] args)
    {
        Employee e1 = new Employee(101,"Aryan");
        Employee e2 = new Employee(102,"Pooja");
        Employee e3 = new Employee(101,"Aryan");
        Employee e4 = e2;

        HashMap<Employee,String> hm = new HashMap<>();
        hm.put(e1,"Ameerpet");
        hm.put(e2,"S.R Nagar");
        hm.put(e3,"Begumpet");
        hm.put(e4,"Panjagutta");
        hm.forEach((k,v)-> System.out.println(k+" : "+v));
    }
}

package com.ravi.hash_map;

import java.util.HashMap;

public class HashMapDemo5
{
    public static void main(String[] args)
    {
        // Create a HashMap to store book titles and their availability (true = available,
false = borrowed)

        HashMap<String, Boolean> library = new HashMap<>();

        library.put("Core Java", true);
        library.put("Advanced Java", true);
        library.put("HTML", false);
        library.put("JavaScript", true);

        // Display the initial library status
        System.out.println("Initial Library Status: ");
        library.forEach((k,v)-> System.out.println(k+ " : "+v));

        // Borrow a book
        String bookToBorrow = "Advanced Java";

        if (library.containsKey(bookToBorrow) && library.get(bookToBorrow))
        {
            library.put(bookToBorrow, false);
            System.out.println(bookToBorrow + " has been borrowed.");
        }
        else
        {
            System.out.println(bookToBorrow + " is not available for borrowing.");
        }

        String bookToReturn = "HTML";

        if (library.containsKey(bookToReturn) && !library.get(bookToReturn))
        {
            library.put(bookToReturn, true); // Mark the book as available
            System.out.println(bookToReturn + "Book has been returned.");
        }
        else
        {
            System.out.println(bookToReturn + "Book is not borrowed.");
        }

        // Check the availability of a book
        String bookToCheck = "JavaScript";

        if (library.containsKey(bookToCheck))
        {
            String availability = library.get(bookToCheck) ? "available" : "borrowed";
            System.out.println(bookToCheck + " Book is " + availability + ".");
        }
        else
        {
            System.out.println(bookToCheck + " is not in the library.");
        }

        //Display the final library status

        System.out.println("Final Library Status:");
        for (HashMap.Entry<String, Boolean> entry : library.entrySet())
        {
            String status = entry.getValue() ? "Available" : "Borrowed";
            System.out.println("Book: " + entry.getKey() + ", Status: " + status);
        }
    }
}

package com.ravi.hash_map;

import java.util.HashMap;

public class HashMapDemo6
{
    public static void main(String[] args)
    {
        String text = "java is fun, java is powerful java is great";

        String[] words = text.split(" ");

        HashMap<String, Integer> wordCount = new HashMap<>();

        for (String word : words)
        {
            wordCount.put(word, wordCount.getDefault(word, 0) + 1);
        }

        System.out.println("Word Frequency: " + wordCount);
    }
}

```