**Home Task#01:**

1. Create a Vector storing integer objects as an input.

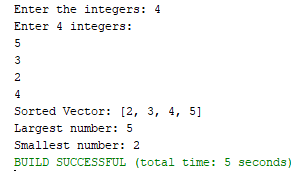
a. Sort the vector

b. Display largest number

c. Display smallest number

**Code: Output:**

import java.util.\*;

public class HomeTask1 {

public static void main(String[] args) {

Vector<Integer> No = new Vector<>();

Scanner no = new Scanner(System.in);

System.out.print("Enter the integers: ");

int n = no.nextInt();

System.out.println("Enter " + n + " integers:");

for (int i = 0; i < n; i++) {

No.add(no.nextInt()); }

Collections.sort(No);

System.out.println("Sorted Vector: " + No);

System.out.println("Largest number: " + No.lastElement());

System.out.println("Smallest number: " + No.firstElement());

}

}

**Home Task#02:**

2. Write a java program which takes user input and gives hashcode value of those inputs using hashCode () method.

**Code:**

import java.util.\*;

public class HomeTask2 {

public static void main(String[] args) {

Scanner inp = new Scanner(System.in);

System.out.println("Enter text for hash code (or 'exit' to quit):");

while (true) {

System.out.print("Text: ");

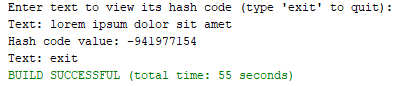
String input = inp.nextLine();

if (input.equals("exit")) break;

System.out.println("Hash code value: " + input.hashCode());}

}

}

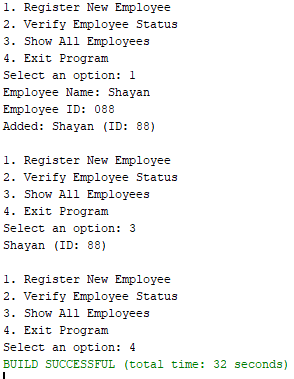
**Output:**

**Home Task#03:**

3. Scenario based Create a java project, suppose you work for a company that needs to manage a list of employees. Each employee has a unique combination of a name and an ID. Your goal is to ensure that you can track employees effectively and avoid duplicate entries in your system.

**Code: Output:**

**Employee.java**

import java.util.\*;

public class Employee {

public String name;

public int id;

public Employee(String name, int id) {

this.name = name;

this.id = id; }

public boolean equals(Object obj) {

if (this == obj) return true;

if (!(obj instanceof Employee)) return false;

Employee e = (Employee) obj;

return id == e.id && Objects.equals(name, e.name); }

public int hashCode() {

return Objects.hash(name, id); }

public String toString() {

return name + " (ID: " + id + ")"; }

}

**EmployeeManagement.java**

import java.util.\*;

public class EmployeeManagement {

public HashSet<Employee> employees = new HashSet<>();

public void addEmployee(String name, int id) {

Employee e = new Employee(name, id);

if (employees.add(e)) {

System.out.println("Added: " + e);

} else {

System.out.println("Already exists: " + e);}

}

public boolean employeeExists(String name, int id) {

return employees.contains(new Employee(name, id));

}

public void displayEmployees() {

if (employees.isEmpty()) {

System.out.println("No employees found.");

} else {

employees.forEach(System.out::println);}

}

public static void main(String[] args) {

Scanner inp = new Scanner(System.in);

EmployeeManagement management = new EmployeeManagement();

while (true) {

System.out.println("\n1. Register New Employee\n2. Verify Employee Status\n3. Show All Employees\n4. Exit Program");

System.out.print("Select an option: ");

int choice = inp.nextInt();

inp.nextLine();

switch (choice) {

case 1:

System.out.print("Employee Name: ");

String name = inp.nextLine();

System.out.print("Employee ID: ");

int id = inp.nextInt();

management.addEmployee(name, id);

break;

case 2:

System.out.print("Employee Name: ");

String checkName = inp.nextLine();

System.out.print("Employee ID: ");

int checkId = inp.nextInt();

System.out.println(management.employeeExists(checkName, checkId) ? "Employee Found" : "Employee Not Found");

break;

case 3:

management.displayEmployees();

break;

case 4:

return;

default:

System.out.println("Invalid selection. Please choose again."); }

}

}

}

**Home Task#04:**

4.Create a Color class that has red, green, and blue values. Two colors are considered equal if their RGB values are the same

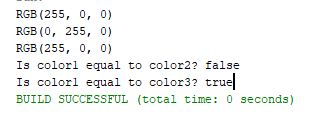
**Code: Output:**

**Color.java**

import java.util.\*;

public class Color {

public int red, green, blue;

 public Color(int red, int green, int blue) {

this.red = clamp(red);

this.green = clamp(green);

this.blue = clamp(blue);

}

public int clamp(int value) {

return Math.max(0, Math.min(value, 255));

}

public boolean equals(Object obj) {

if (this == obj) return true;

if (!(obj instanceof Color)) return false;

Color color = (Color) obj;

return red == color.red && green == color.green && blue == color.blue;

}

public int hashCode() {

return Objects.hash(red, green, blue);

}

public String toString() {

return "RGB(" + red + ", " + green + ", " + blue + ")";

}

}

**Main.java**

public class Main {

public static void main(String[] args) {

Color c1 = new Color(255, 0, 0);

Color c2 = new Color(0, 255, 0);

Color c3 = new Color(255, 0, 0);

System.out.println(c1);

System.out.println(c2);

System.out.println(c3);

System.out.println("Is color1 equal to color2? "+c1.equals(c2));

System.out.println("Is color1 equal to color3? "+c1.equals(c3));

}

}