**Program 1:**

package Hashmap;

import java.util.ArrayList;

public class GenericList<T>

{

private ArrayList<T> elements = new ArrayList<>();

public void addElement(T element)

{

elements.add(element);

}

public T getElement(int index)

{

return elements.get(index);

}

public void printList()

{

for (T element : elements)

{

System.out.println(element);

}

}

}

class GenericAccess

{

public static void main(String[] args)

{

// creating object of GenericList class

GenericList<String> stringList = new GenericList<>();

// adding element

stringList.addElement("Apple");

stringList.addElement("Banana");

stringList.addElement("Cherry");

//Printing List

stringList.printList();

// accessing by it's index

System.out.println("Element at index 1: " + stringList.getElement(1));

}

}

**Output:**

Apple

Banana

Cherry

Element at index 1: Banana

**Program 2:**

package Hashmap;

public class MaxElementFinder

{

public static <T extends Comparable<T>> T findMaxElement(T[] array)

{

if (array == null || array.length == 0)

{

return null;

}

T max = array[0];

for (int i = 1; i < array.length; i++)

{

if (array[i].compareTo(max) > 0)

{

max = array[i];

}

}

return max;

}

public static void main(String[] args)

{

//create and adding arrays

Integer[] intArray = {1, 5, 3, 9, 2};

Double[] doubleArray = {2.5, 1.2, 3.7, 2.0};

String[] stringArray = {"apple", "banana", "orange", "kiwi","grape"};

// printing maximum

System.out.println("Maximum Integer: " + findMaxElement(intArray));

System.out.println("Maximum Double: " + findMaxElement(doubleArray));

System.out.println("Maximum String: " + findMaxElement(stringArray));

}

}

**Output**

Maximum Integer: 9

Maximum Double: 3.7

Maximum String: orange

**Program 3:**

package Hashmap;

import java.util.HashSet;

class Employee {

private int employeeID;

private String employeeName;

public Employee(int employeeID, String employeeName) {

this.employeeID = employeeID;

this.employeeName = employeeName;

}

@Override

public boolean equals(Object obj) {

if (this == obj) {

return true;

}

if (obj == null || getClass() != obj.getClass()) {

return false;

}

Employee employee = (Employee) obj;

return employeeID == employee.employeeID;

}

public static void main(String[] args) {

Employee emp1 = new Employee(1, "Alice");

Employee emp2 = new Employee(2, "Bob");

Employee emp3 = new Employee(3, "Charlie");

HashSet<Employee> employeeSet = new HashSet<>();

employeeSet.add(emp1);

employeeSet.add(emp2);

employeeSet.add(emp3);

Employee emp4 = new Employee(1, "David");

employeeSet.add(emp4);

for (Employee emp : employeeSet) {

System.out.println("Employee ID: " + emp.employeeID + ", Employee Name: " + emp.employeeName);

}

}

}

**Output:**

Employee ID: 1, Employee Name: Alice

Employee ID: 2, Employee Name: Bob

Employee ID: 3, Employee Name: Charlie

Employee ID: 1, Employee Name: David

**Program 4:**

package Hashmap;

import java.util.HashSet;

import java.util.Objects;

import java.util.Set;

class Student {

private String studentName;

private int studentID;

public Student(String studentName, int studentID)

{

this.studentName = studentName;

this.studentID = studentID;

}

@Override

public boolean equals(Object o)

{

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

Student student = (Student) o;

return studentID == student.studentID;

}

@Override

public int hashCode()

{

return Objects.hash(studentID);

}

}

public class Main

{

public static void main(String[] args)

{

Set<Student> studentSet = new HashSet<>();

Student student1 = new Student("Alice", 1);

Student student2 = new Student("Bob", 2);

Student student3 = new Student("Charlie", 3);

// adding Students to student set

studentSet.add(student1);

studentSet.add(student2);

studentSet.add(student3);

System.out.println(studentSet);

Student student4 = new Student("David", 1);

// Same studentID as student1

boolean isAdded = studentSet.add(student4);

// checking new student is added or not

System.out.println("Is the new student added to the set? " + isAdded);

System.out.println(studentSet);

}

}

**Output:**

[Hashmap.Student@20, Hashmap.Student@21, Hashmap.Student@22]

Is the new student added to the set? false

[Hashmap.Student@20, Hashmap.Student@21, Hashmap.Student@22]