Course Name:	**Object Oriented Programming Methodology**	**Semester:**	**III**
Date of Performance:	**22 /09/ 2023**	**Batch No:**	**B-2**
Faculty Name:	**Prof. Kiran Thale**	**Roll No:**	**16010122151**
Faculty Sign & Date:		**Grade/Marks:**	**___/25**

Experiment No: 5
Title: Vector

Aim and Objective of the Experiment:

Create a class Employee which stores E-Name, E-Id and E-Salary of an Employee. Use class Vector to maintain an array of Employee with respect to the E-Salary. Provide the following functions

1) Create (): this function will accept the n Employee records in any order and will arrange them in the sorted order.

2) Insert (): to insert the given Employee record at appropriate index in the vector depending upon the E-Salary.

3) delete ByE-name( ): to accept the name of the Employee and delete the record having given name

4) deleteByE-Id ( ): to accept the Id of the Employee and delete the record having given E-Id.

Provide the following functions

1. boolean add(E e) : This method appends the specified element to the end of this Vector.
2. void addElement(E obj) This method adds the specified component to the end of this vector, increasing its size by one.
3. int lastIndexOf(Object o, int index) This method returns the index of the last occurrence of the specified element in this vector, searching backwards from index, or returns -1 if the element is not found.
4. void removeElementAt(int index)This method deletes the component at the specified index.

COs to be achieved:

CO2: Explore arrays, vectors, classes and objects in C++ and Java.

Tools used:

JDK, VScode / Eclipse

Theory:

Pre Lab/ Prior Concepts:

Vectors in Java are one of the most commonly used data structures. Similar to Arrays data structures which hold the data in a linear fashion. Vectors also store the data in a linear fashion, but unlike Arrays, they do not have a fixed size. Instead, their size can be increased on demand.
Vector class is a child class of AbstractList class and implements on List interface. To use Vectors, we first have to import Vector class from java.util package:
import java.util.Vector;

Access Elements in Vector:

We can access the data members simply by using the index of the element, just like we access the elements in Arrays.
Example- If we want to access the third element in a vector v, we simply refer to it as v[3].

Vectors Constructors

Listed below are the multiple variations of vector constructors available to use:

1. **Vector(int initialCapacity, int Increment)** – Constructs a vector with given initialCapacity and its Increment in size.
2. **Vector(int initialCapacity) –** Constructs an empty vector with given initialCapacity. In this case, Increment is zero.
3. **Vector()** – Constructs a default vector of capacity 10.
4. **Vector(Collection c) –** Constructs a vector with a given collection, the order of the elements is same as returned by the collection's iterator.

There are also three protected parameters in vectors

- **Int capacityIncrement()-** It automatically increases the capacity of the vector when the size becomes greater than capacity.
- **Int elementCount()** – tell number of elements in the vector
- **Object[] elementData()** – array in which elements of vector are stored

Memory allocation of vectors:

Vectors do not have a fixed size, instead, they have the ability to change their size dynamically. One might think that the vectors allocate indefinite long space to store objects. But this is not the case. Vectors can change their size based on two fields 'capacity' and 'capacityIncrement'. Initially, a size equal to 'capacity' field is allocated when a vector is declared. We can insert the elements equal to the capacity. But as soon as the next element is inserted, it increases the size of the array by size 'capacityIncrement'. Hence, it is able to change its size dynamically.
For a default constructor, the capacity is doubled whenever the capacity is full and a new element is to be inserted.

Methods of Vectors :

- Adding elements
- Removing elements
- Changing elements
- Iterating the vector

Class Diagram:

Class name	vector
variables	int count
object	Vector<Employee>
Function	Main, create, sort, insert, dename, delid,display

Class name	Employee
variables	double E_salary ,string E_name, int E_Id
Function	Employee

Algorithm:

1. Start the program.
2. The control first goes to the main() function. It creates the vector of Employee structure.
3. Now a will loop runs until user doesn't wants to exit the program and it asks user to input in from 0-5.
4. If user enters 1 than it calls the addElelemt() function. The function ask user no. of user details to be entered and if the vector is not full then it creates the vector and then the user can enter the required employee credential
5. If user enters 2 than it checks whether the vector is not full than it ask user to enter the required data and adds employee using add method in function.
6. If user enters 3 then it checks whether the vector is empty or not, if not than it ask user the name to be deleted and it removes the employee using

remove function.
7. If user enters 4 then it checks whether the vector is empty or not, if not
than it ask user the id to be deleted and it removes the employee using
remove function.
8. If user enter 5 then it prints the all the employee details in the vector.
9. If user enter 0 the it exits the program.

Output:

```
***Main Menu***
1. Create employee records.
2. Insert employee record according to their salary.
3. Delete employee record according to name.
4. Delete employee record according to id.
5. Display the list.
6. Exit.
Enter your choice:
1
Enter the total number of employee records you want to add:
2
Details of employee 1:
Enter id of the employee:
123
Enter employee name:
Youv
Enter salary of the employee:
100
Details of employee 2:
Enter id of the employee:
1234
Enter employee name:
Yoyo
Enter salary of the employee:
200
1. Create employee records.
2. Insert employee record according to their salary.
3. Delete employee record according to name.
4. Delete employee record according to id.
5. Display the list.
6. Exit.
```

```
1. Create employee records.
2. Insert employee record according to their salary.
3. Delete employee record according to name.
4. Delete employee record according to id.
5. Display the list.
6. Exit.
Enter your choice:
5
Employee 1:
Name: Youz
ID: 123
Salary: 100.0
Employee 2:
Name: 20Yoyo
ID: 1234
Salary: 200.0
1. Create employee records.
2. Insert employee record according to their salary.
3. Delete employee record according to name.
4. Delete employee record according to id.
5. Display the list.
6. Exit.
```

Post Lab Subjective/Objective type Questions:

What is the output of the following code ?

```
import java.util.*;
class demo2
{
    public static void main(String[] args)
    {
        Vector v = new Vector(20); v.addElement("Geeksforgeeks");
        v.insertElementAt("Java", 2); System.out.println(v.firstElement());
    }
}
```

Output: Geeksforgeeks

2) **Expain any 10 methods of Vector class in detail with the help of example**

1. add(E element): Adds the specified element to the end of the vector.
2. capacity(): Gets the current capacity of this vector.
3. clear(): Deletes all elements from the vector.
4. contains(Object o): Returns true if the vector contains the specified element.
5. indexOf(Object o): Gets the index of the first occurrence of the specified element in the vector or returns -1 if not found.
6. insertElementAt(E obj, int index): Inserts the specified object at the specified index in the vector.
7. elementAt(int index): Gets the component at the specified index.
8. isEmpty(): Checks if the vector has no components.
9. remove(Object o): Removes the specified element from the vector. If the vector does not contain the element, it remains unchanged.
10. sort(Comparator<? super E> c): Sorts the list according to the order induced by the speci

Conclusion:

This experiment helps in understanding the fundamental concepts of working with vectors of objects in Java, which can be extended to more complex applications for handling various data structures and collections.

Signature of faculty in-charge with Date: