Batch: D2	**Roll No.: 16010122323**
Experiment / assignment / tutorial No.05 **Grade: AA / AB / BB / BC / CC / CD /DD**	
Signature of the Staff In-charge with date	

TITLE :Vector

AIM: 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 ESalary. 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.

Expected OUTCOME of Experiment:

CO2: Solve problems using Java basic constructs (like if else statement, control structures, and data types, array, string, vectors, packages, collection class).

Books/ Journals/ Websites referred:

1. Ralph Bravaco , Shai Simoson , “Java Programing From the Group Up” Tata McGraw-Hill.

2. Grady Booch, Object Oriented Analysis and Design .

Pre Lab/ Prior Concepts:

Class Diagram:

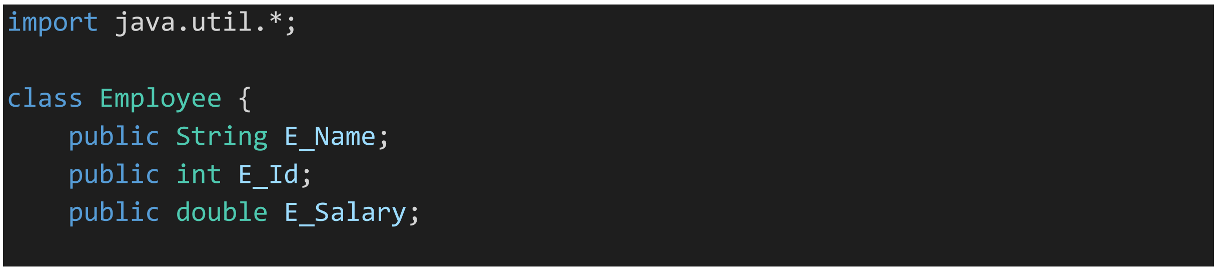
Variables	Choice, n, i, e_id, e_salary, e_name, emp →vector
Class	Employee
Objects	s
Functions	addElement(), add(), size(), get(), remove(), equals(), exit(), main()

Algorithm:

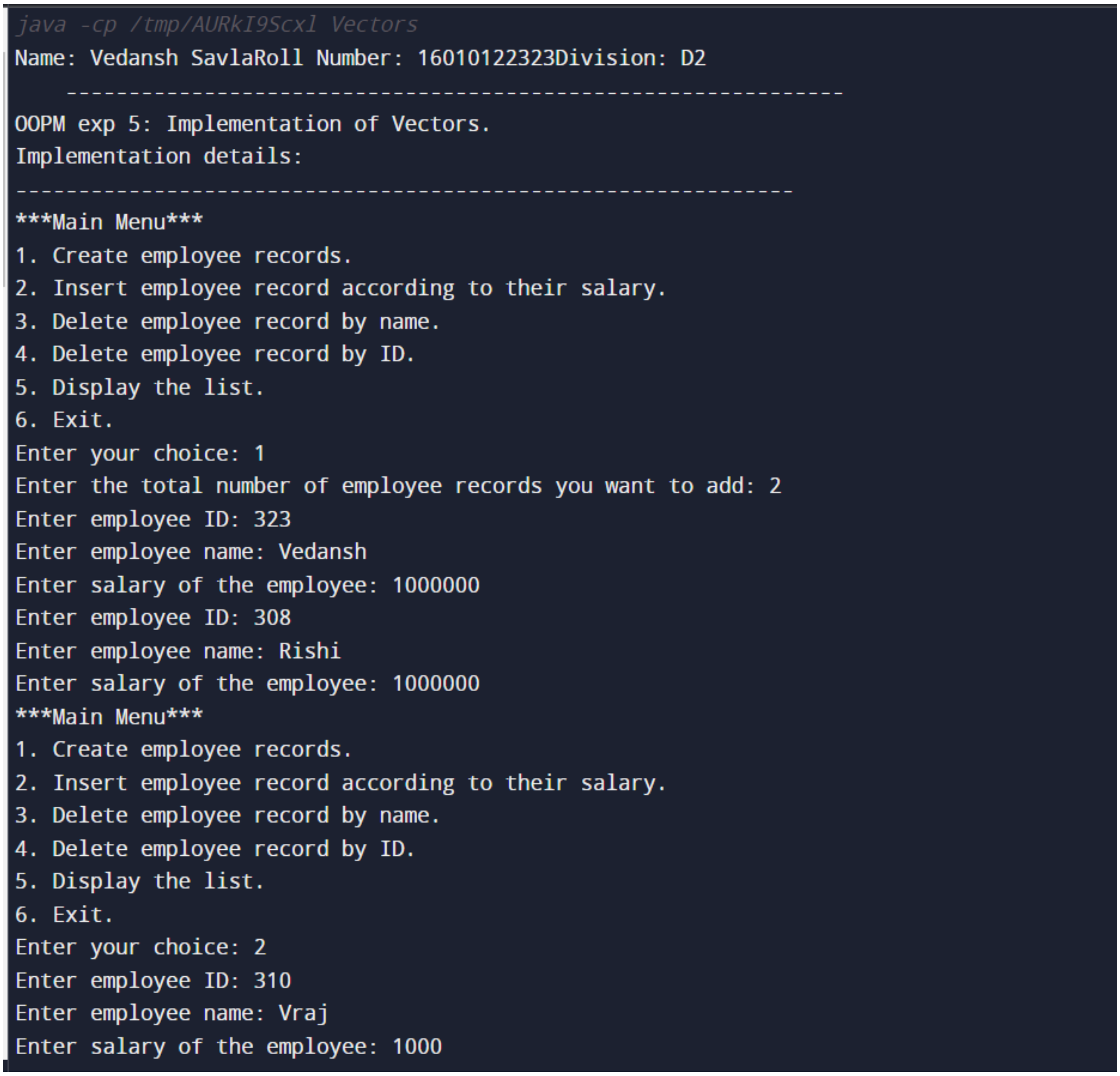
1. Start
2. Create a Vector object Emp of the type Employee(E_Name, E_Id, E_Salary)

```
3.  while(true)
4.  1.Create  2.Insert by salary 3.Delete by name 4.Delete by Id  5.Display  6. Exit
5.  read choice
6.  (Switch Case), Case 1(choice=1)
      6.1 read number of employee records to be added(n)
      6.2 for i=0, i<n, accept the employee details(e_name. e_id, e_salary)
      6.3 Sort the employee records using comparator interface
            6.3.1 if a.E_Salary>b.E_Salary, return 1
            6.3.2 else if a.E_Salary<b.E_Salary, return -1
            6.3.3 else, return 0
7.  Case 2(choice=2)
      7.1 accept the employee details
      7.2 for i=0, i<Emp.size()
      If Emp.get(i).E_Salary>e_salary, add the record to the ith index.
8.  Case 3(choice=3)
      8.1 Declare a=0
      8.2 read the name of record to be deleted          8.3 for i=0, i<Emp.size()
      if e_name = Emp.get(i).E_Name, remove the corresponding record and
increment a
    8.4 if a=0, print "Employee name not found."
9.  Case 4(choice=4)
      9.1 Declare b=0
      9.2 read the name of record to be deleted          9.3 for i=0, i<Emp.size()
         if e_id = Emp.get(i).E_Id, remove the corresponding record and increment
         b

    9.4 if b=0, print "Employee id not found."
10. Case 5(choice=5)
    Print all the records
11. Case 6(choice=6)
    Exit
```

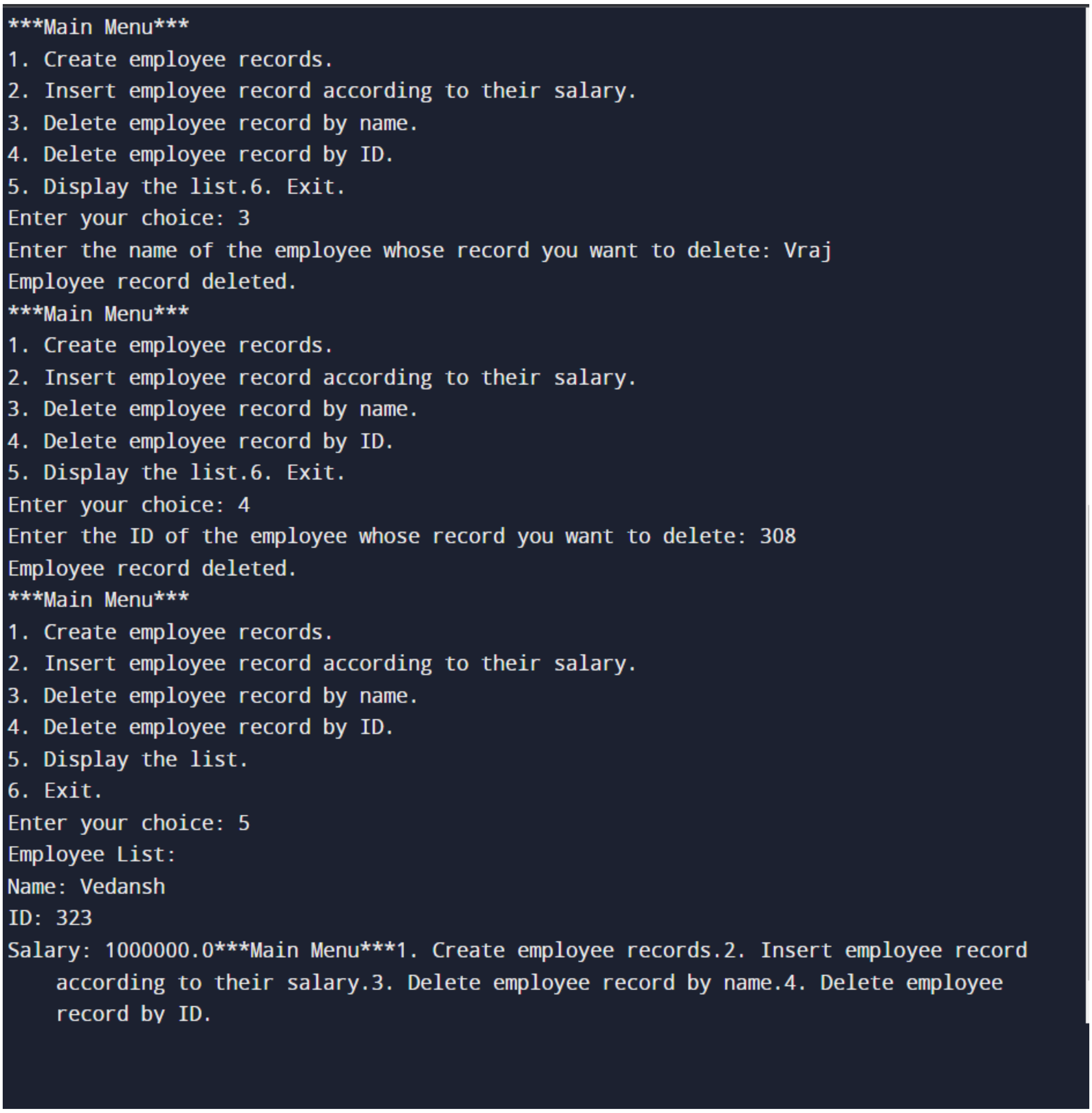
Implementation details: (printout of code):



```java
import java.util.*;

class Employee {
    public String E_Name;
    public int E_Id;
    public double E_Salary;

    Employee(String name, int id, double salary) {
        E_Name = name;
        E_Id = id;
        E_Salary = salary;
    }
}

class SortBySalary implements Comparator<Employee> {
    public int compare(Employee a, Employee b) {
        return Double.compare(a.E_Salary, b.E_Salary);
    }
}

class Vectors {
    public static void main(String[] args) {
        System.out.println("Name: Vedansh Savla");
        System.out.println("Roll Number: 16010122323");
        System.out.println("Division: D2");
        System.out.println("--------------------------------------------------
------------");
        System.out.println("OOPM exp 5: Implementation of Vectors.");
        System.out.println("Implementation details:");
        System.out.println("--------------------------------------------------
------------");

        Scanner scanner = new Scanner(System.in);
        Vector<Employee> employees = new Vector<>();

        while (true) {
            System.out.println("***Main Menu***");
            System.out.println("1. Create employee records.");
            System.out.println("2. Insert employee record according to their
salary.");
            System.out.println("3. Delete employee record by name.");
            System.out.println("4. Delete employee record by ID.");
            System.out.println("5. Display the list.");
            System.out.println("6. Exit.");

            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter the total number of employee
records you want to add: ");
                    int n = scanner.nextInt();
                    for (int i = 0; i < n; i++) {
                        System.out.print("Enter employee ID: ");
                        int e_id = scanner.nextInt();
                        System.out.print("Enter employee name: ");
                        String e_name = scanner.next();
                        System.out.print("Enter salary of the employee: ");
                        double e_salary = scanner.nextDouble();
                        employees.add(new Employee(e_name, e_id, e_salary));
                    }
                    Collections.sort(employees, new SortBySalary());
                    break;

                case 2:
                    System.out.print("Enter employee ID: ");
                    int e_id = scanner.nextInt();
                    System.out.print("Enter employee name: ");
                    String e_name = scanner.next();
                    System.out.print("Enter salary of the employee: ");
                    double e_salary = scanner.nextDouble();

                    for (int i = 0; i < employees.size(); i++) {
                        if (employees.get(i).E_Salary > e_salary) {
                            employees.add(i, new Employee(e_name, e_id,
e_salary));

                            break;
                        }
                    }
                    break;

                case 3:
                    System.out.print("Enter the name of the employee whose
record you want to delete: ");
                    String deleteName = scanner.next();
                    Iterator<Employee> iteratorByName = employees.iterator();
                    while (iteratorByName.hasNext()) {
                        Employee employee = iteratorByName.next();
                        if (deleteName.equals(employee.E_Name)) {
                            iteratorByName.remove();
                            System.out.println("Employee record deleted.");
                            break;
                        }
                    }
                    break;

                case 4:
                    System.out.print("Enter the ID of the employee whose
record you want to delete: ");
                    int deleteId = scanner.nextInt();
                    Iterator<Employee> iteratorById = employees.iterator();
                    while (iteratorById.hasNext()) {
                        Employee employee = iteratorById.next();
                        if (employee.E_Id == deleteId) {
                            iteratorById.remove();
                            System.out.println("Employee record deleted.");
                            break;
                        }
                    }
                    break;

                case 5:
                    System.out.println("Employee List:");
                    for (Employee emp : employees) {
                        System.out.println("Name: " + emp.E_Name);
                        System.out.println("ID: " + emp.E_Id);
                        System.out.println("Salary: " + emp.E_Salary);
                        System.out.println();
                    }
                    break;

                case 6:
                    System.out.println("EXIT");
                    System.exit(0);
                    break;

                default:
                    System.out.println("Invalid choice. Please enter a valid
option.");

                    break;
            }
        }
    }
}
```



```
java -cp /tmp/AURkI9Scxl Vectors
Name: Vedansh SavlaRoll Number: 16010122323Division: D2
    -------------------------------------------------------------
OOPM exp 5: Implementation of Vectors.
Implementation details:
-------------------------------------------------------------
***Main Menu***
1. Create employee records.
2. Insert employee record according to their salary.
3. Delete employee record by name.
4. Delete employee record by ID.
5. Display the list.
6. Exit.
Enter your choice: 1
Enter the total number of employee records you want to add: 2
Enter employee ID: 323
Enter employee name: Vedansh
Enter salary of the employee: 1000000
Enter employee ID: 308
Enter employee name: Rishi
Enter salary of the employee: 1000000
***Main Menu***
1. Create employee records.
2. Insert employee record according to their salary.
3. Delete employee record by name.
4. Delete employee record by ID.
5. Display the list.
6. Exit.
Enter your choice: 2
Enter employee ID: 310
Enter employee name: Vraj
Enter salary of the employee: 1000
```



```
***Main Menu***
1. Create employee records.
2. Insert employee record according to their salary.
3. Delete employee record by name.
4. Delete employee record by ID.
5. Display the list.6. Exit.
Enter your choice: 3
Enter the name of the employee whose record you want to delete: Vraj
Employee record deleted.
***Main Menu***
1. Create employee records.
2. Insert employee record according to their salary.
3. Delete employee record by name.
4. Delete employee record by ID.
5. Display the list.6. Exit.
Enter your choice: 4
Enter the ID of the employee whose record you want to delete: 308
Employee record deleted.
***Main Menu***
1. Create employee records.
2. Insert employee record according to their salary.
3. Delete employee record by name.
4. Delete employee record by ID.
5. Display the list.
6. Exit.
Enter your choice: 5
Employee List:
Name: Vedansh
ID: 323
Salary: 1000000.0***Main Menu***1. Create employee records.2. Insert employee record
    according to their salary.3. Delete employee record by name.4. Delete employee
    record by ID.
```

Conclusion: The concept of vectors in Java was studied and implemented.

Post Lab Descriptive Questions (Add questions from examination point view)

1) What is the Output Of the following Program?

```
import java.util.*; classdemo2 {
    public static void main(String[] args)
    {
      Vector v = new Vector(20);
      v.addElement("Geeksforgeeks");
      v.insertElementAt("Java",                              2);
  System.out.println(v.firstElement());
    }
  }
```

ANSWER: It gives no output because of ArrayindexOutOfBound Exception because initially there was only one element at index 0. Next it expects a value to be added at index 1, but we are adding at index 2. Hence an exception is raised.

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

ANSWER:

1. **void add(int index, Object element)**

   Inserts the specified element at the specified position in this Vector.

   Example: v.add(1, “apple”);

2. **boolean add(Object o)**

   Appends the specified element to the end of this Vector.

   Example: v.add(“pear”);

3. **boolean addAll(int index, Collection c)**

   Inserts all of the elements in the specified Collection into this Vector at the specified position.

   Example: v.addAll(3, arr);

4. **void addElement(Object obj)**

   Adds the specified component to the end of this vector, increasing its size by one.

Example: v.addElement(“plum”);

5. **int capacity()**

   Returns the current capacity of this vector.

   Example: v.capacity();

6. **boolean contains(Object elem)**

   Tests if the specified object is a component in this vector.

   Example: v.contains(“apple”);

7. **void copyInto(Object[] anArray)**

   Copies the components of this vector into the specified array.

   Example: v.copyInto(arr);

8. **Object elementAt(int index)**

   Returns the component at the specified index.

   Example: v.elementAt(2);

9. **Object remove(int index)**

   Removes the element at the specified position in this vector.

   Example: v. remove(3);

10. **int size()**

    Returns the number of components in this vector.

    Example: v.size();

Date: 8th Oct 2023 **Signature of faculty in-charge**