



**K. J. Somaiya College of Engineering, Mumbai-77**

**Batch: A3      Roll No.: 1911034**

**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 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.



## K. J. Somaiya College of Engineering, Mumbai-77

### 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:

Class:

public Class Employee()

#### Methods

1. public static void main(String args[])
2. public static Vector<Employee> Create(Vector<Employee> v, int n)//method to initially create the list of employees to display them in sorted order
3. public static Vector<Employee> Insert(Vector<Employee> v)//method to insert Employee record sorted according to salary
4. public static void Display(Vector<Employee> v)
5. public static Vector<Employee> deletebyId(Vector<Employee> v)
6. public static Vector<Employee> deletebyName(Vector<Employee> v)

#### Variables :

Class instance variables : String name , int id , float sal;

Inside main():int n,ch,ch1;

Inside Create():String ename;

float esal;

int eid;

int i,j;



## K. J. Somaiya College of Engineering, Mumbai-77

```
Inside Insert(): String name1; int id1,i; float sal1;  
In Display(): int i, Enumeration en;  
In deletebyID: int i,num;  
In deletebyName(): String dname;  
int i,j,k1=0,k2=0,flag=0,t_id,k;
```

### **Algorithm:**

1. Create a Class Employee , having the main method().
2. Declare three Instance Variables name , id and salary belonging to Employee objects
3. Inside the main method , declare a vector 'vec' of type Class Employee, to store employee objects.
4. By asking user for initial number of accounts to be created (say n), call the Create() method n times, by passing the vector, as an argument.
5. Repeat steps 6 to 8 for i=1 to i=n in Create()
6. Create three variables , ename , eid and esal belonging to create.
7. Create an object obj of type Employee , and by asking user for these values , assign them to the object.
8. Using v.addElement() method , add this object to the vector
9. By creating a temporary variable , temp of type Employee Class; sort the elements of Employee sorted in increasing order of salary , by using the 'dot' operator to access the salary field of the objects, using bubble sort
10. Display the elements in increasing order of the salary of each employee.
11. In the method Insert(), create an object of type 'Employee', and by asking user for the three fields , insert the new record at the appropriate position sorted in



## K. J. Somaiya College of Engineering, Mumbai-77

increasing order of salary (by using a for loop to access the elements, salary) using the `insertElementAt()` method

12. In the method `deletebyID`, create an integer variable to accept the user ID to be deleted and compare it with all the IDs of the objects of the vector using dot operator. If the ID is found, delete the record containing the ID, using `removeElementAt(i)`, where *i* is the index of the Employee with the ID.
13. In the method `deletebyName()` create a String variable for to accept name of user whose account is to be deleted. In case 2 accounts with the same name are found, display the ID number of the two accounts and ask user for the ID to be deleted.
14. Again using `removeElementAt ()` method, the required record of the employee can be deleted.
15. We can also display the elements by calling the `Display()` method.
16. In the display method, create a variable of Enumeration interface, for traversing through all the elements in the vector until `enum.hasMoreElements()` returns true.
17. Using switch case statement in `main()`, all the modules to insert, create, display or delete can be accessed.

### **Implementation details:** (printout of code)

Code :

```
import java.util.*;
public class Employee
{ String name;
  float sal;
  int id;

  public static void main(String args[])
  {
    Vector<Employee> vec = new Vector<Employee> (); //creating a vector which
    will store objects of class Employee.
    int n,ch,ch1;
    System.out.println("Enter the number of employees");
    Scanner sc = new Scanner(System.in);
```



## K. J. Somaiya College of Engineering, Mumbai-77

```
n = sc.nextInt();  
vec=Create(vec,n);
```

```
do  
{
```

```
    System.out.println("Enter any 1 of the following choices ");  
    System.out.println("1 to insert a new record");  
    System.out.println("2 to delete an Employee record by name");  
    System.out.println("3 to delete by the ID");  
    System.out.println("4 to display all Employee Records sorted by salary");  
    System.out.println("-1 to exit");  
    ch = sc.nextInt();
```

```
    switch(ch)  
    {  
        case 1:  
            {  
                vec = Insert(vec);  
            }  
        break;  
        case 2:  
            {  
                vec = deletebyName(vec);  
            }  
        break;  
  
        case 3:  
            {  
                vec = deletebyId(vec);  
            }  
        break;  
  
        case 4:  
            {  
  
                Display(vec);  
            }  
    }
```



## K. J. Somaiya College of Engineering, Mumbai-77

```
    }

}

}while(ch!=-1);

}

public static Vector<Employee> Create(Vector<Employee> v, int n)//this
method will be used to create the initial list of n Employees sorted
according to salary
{
    String ename;
    float esal;
    int eid;
    int i,j;
    Employee temp = new Employee();//calling default constructor to
initialize temporary variable.
    Scanner sc1 = new Scanner(System.in);
    for(i=0;i<n;i++)
    {
        System.out.println("Enter the ID");
        eid = sc1.nextInt();
        System.out.println("Enter the name");
        ename = sc1.next();
        System.out.println("Enter the salary");
        esal = sc1.nextFloat();
        System.out.println("-----");
        Employee obj = new Employee();
        obj.name = ename;
        obj.sal = esal;
        obj.id = eid;
        v.addElement(obj);
    }

    for(i=0;i<n;i++)//to sort the Employees in increasing order of salary
    {
        for(j=0;j<n-i-1;j++)
```



## K. J. Somaiya College of Engineering, Mumbai-77

```
{
if(v.elementAt(j).sal>v.elementAt(j+1).sal)
{
temp = v.elementAt(j);
v.set(j , v.elementAt(j+1));
v.set(j+1, temp);

}

}
}

System.out.println("The sorted list of Employees according to the salary
is");
for(i=0;i<n;i++)
{
System.out.println("ID :" +v.elementAt(i).id);
System.out.println("Name :" +v.elementAt(i).name);
System.out.println("Salary :" +v.elementAt(i).sal);
System.out.println("-----");

}

return v;

}

public static Vector<Employee> Insert(Vector<Employee> v)//method to
insert Employee record sorted according to salary
{
Employee obj1 = new Employee();
String name1; int id1,i; float sal1;
Scanner sc1 = new Scanner(System.in);
System.out.println("Enter the id");
id1 = sc1.nextInt();
obj1.id = id1;
System.out.println("Enter the name");
name1 = sc1.next();
obj1.name = name1;
System.out.println("Enter the Salary");
sal1 = sc1.nextFloat();
obj1.sal = sal1;
```



## K. J. Somaiya College of Engineering, Mumbai-77

```
for(i=0;i<v.size();i++)
{
if(v.elementAt(i).sal> sal1)
break;

}

v.insertElementAt(obj1 , i);
return v;

}

public static void Display(Vector<Employee> v)

{
int i;
Enumeration<Employee> en = v.elements();// Enum is used to iterate
through the elements of the vector.

while(en.hasMoreElements())
{
Employee obj1 = new Employee();
obj1 = en.nextElement();
System.out.println("ID :" +obj1.id);
System.out.println("Name :" +obj1.name);
System.out.println("Salary :" +obj1.sal);
System.out.println("-----");
}

}

public static Vector<Employee> deletebyId(Vector<Employee> v)

{
int i,num;
Scanner sc = new Scanner(System.in);
System.out.println("Enter the ID of the Employee you wish to remove");
num = sc.nextInt();
for(i=0;i<v.size();i++)
{
```





## K. J. Somaiya College of Engineering, Mumbai-77

```
if(v.elementAt(i).id == num)// to search for the record having the
required ID.
{
break;
}

}
v.removeElementAt(i);
System.out.println("Employee Record deleted sucesssfully");
return v;

}

public static Vector<Employee> deletebyName(Vector<Employee> v)
{
String dname;
int i,j,k1=0,k2=0,flag=0,t_id,k;

Scanner sc = new Scanner(System.in);
System.out.println("Enter the name of the Employee whose record you wish
to delete");
dname = sc.next();
for(i=0;i<v.size();i++)
{
if(v.elementAt(i).name.equals(dname)== true)
{
k1 = v.elementAt(i).id;
break;

}
}
for(j=i+1;j<v.size();j++)// loop used to check whether two employees with
the same name but different ID's exist.
{
if(v.elementAt(j).name.equals(dname)== true)
{
k2 = v.elementAt(j).id;
flag =1;
break;

}
}
```



## K. J. Somaiya College of Engineering, Mumbai-77

```
}

if (flag==0)
{
v.removeElementAt(i);

}
else if(flag==1)
{
System.out.println("There are two employees with the same name and with
the ID's" +k1+ " and " +k2+ " are found. Enter the ID of the Employee
you want to delete");

t_id = sc.nextInt();

for(i=0;i<v.size();i++)
{
if(v.elementAt(i).id==t_id)
{
v.removeElementAt(i);

}

}

}
System.out.println("Employee record deleted successfully");

return v;

}

}
```

**Output :**



## K. J. Somaiya College of Engineering, Mumbai-77

Case 1 :

Adding a record containing 5 Employees , which then displays them in sorted order (according to salary): :

```
C:\Users\Aditi Paretkar\Desktop\OOPM JAVA programs>java Employee
Enter the number of employees
5
Enter the ID
101
Enter the name
Aditi
Enter the salary
1500
-----
Enter the ID
102
Enter the name
Arvind
Enter the salary
2100
-----
Enter the ID
103
Enter the name
Rahul
Enter the salary
1700
-----
Enter the ID
104
Enter the name
Aditi
Enter the salary
1100
-----
Enter the ID
105
Enter the name
Prachi
Enter the salary
1900
-----
The sorted list of Employees according to the salary is
```



## K. J. Somaiya College of Engineering, Mumbai-77

```
-----
The sorted list of Employees according to the salary is
ID :104
Name :Aditi
Salary :1100.0
-----
ID :101
Name :Aditi
Salary :1500.0
-----
ID :103
Name :Rahul
Salary :1700.0
-----
ID :105
Name :Prachi
Salary :1900.0
-----
ID :102
Name :Arvind
Salary :2100.0
-----
Enter any 1 of the following choices
1 to insert a new record
2 to delete an Employee record by name
3 to delete by the ID
4 to display all Employee Records sorted by salary
-1 to exit
```

Case 2: Adding a new Employee at the appropriate position sorted according to salary, and displaying(use Insert and Display function) :

```
Enter the id
106
Enter the name
Ashwini
Enter the Salary
2050
Enter any 1 of the following choices
1 to insert a new record
2 to delete an Employee record by name
3 to delete by the ID
4 to display all Employee Records sorted by salary
-1 to exit
4
ID :104
Name :Aditi
Salary :1100.0
-----
ID :101
Name :Aditi
Salary :1500.0
-----
ID :103
Name :Rahul
Salary :1700.0
-----
ID :105
Name :Prachi
Salary :1900.0
-----
ID :106
Name :Ashwini
Salary :2050.0
-----
ID :102
Name :Arvind
Salary :2100.0
-----
```



## K. J. Somaiya College of Engineering, Mumbai-77

Case 3: Deleting an Employee record by ID then displaying full list again after deleting

```
Enter the ID of the Employee you wish to remove
103
Employee Record deleted sucesssfully
Enter any 1 of the following choices
1 to insert a new record
2 to delete an Employee record by name
3 to delete by the ID
4 to display all Employee Records sorted by salary
-1 to exit
4
ID :104
Name :Aditi
Salary :1100.0
-----
ID :101
Name :Aditi
Salary :1500.0
-----
ID :105
Name :Prachi
Salary :1900.0
-----
ID :106
Name :Ashwini
Salary :2050.0
-----
ID :102
Name :Arvind
Salary :2100.0
```

Case 4 : Deleting an Employee Record by name , then displaying :



## K. J. Somaiya College of Engineering, Mumbai-77

```
Enter the name of the Employee whose record you wish to delete
Prachi
Employee record deleted successfully
Enter any 1 of the following choices
1 to insert a new record
2 to delete an Employee record by name
3 to delete by the ID
4 to display all Employee Records sorted by salary
-1 to exit
4
ID :104
Name :Aditi
Salary :1100.0
-----
ID :101
Name :Aditi
Salary :1500.0
-----
ID :106
Name :Ashwini
Salary :2050.0
-----
ID :102
Name :Arvind
Salary :2100.0
-----
```

Case 5: Here we have 2 Employees with name 'Aditi', thus when we try to delete by the Name, it will ask us for the ID to find the specific account to be deleted :

```
2
Enter the name of the Employee whose record you wish to delete
Aditi
There are two employees with the same name and with the ID's 104 and 101 are found. Enter the ID of the Employee you want to delete
101
Employee record deleted successfully
Enter any 1 of the following choices
1 to insert a new record
2 to delete an Employee record by name
3 to delete by the ID
4 to display all Employee Records sorted by salary
-1 to exit
4
ID :104
Name :Aditi
Salary :1100.0
-----
ID :106
Name :Ashwini
Salary :2050.0
-----
ID :102
Name :Arvind
Salary :2100.0
-----
```



## K. J. Somaiya College of Engineering, Mumbai-77

### Conclusion

In this experiment , we have learnt about the Vector Class, and we have create a Vector Class to store objects having 3 fields , namely ID , Name and Salary. Using various methods of Vector Class, we have successfully sorted the list and well as inserted the Employee record at the appropriate position. We have also used the delete methods of Vector Class, to delete the particular Employee Record according to Name or ID.

Date:\_\_\_\_\_

Signature of faculty in-charge

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

#### 1) What is the Output Of the following Program

```
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());
    }
}
```

**Ans:** It will give no output but 'ArrayIndexOutOfBoundsException' Exception , since the first Element 'GeeksforGeeks' is stored at index 0 , thus it expects next element to be stored at index 1, so accessing index 2 to store 'Java' will cause this exception

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



The methods of Vector class are as follows

1. **boolean add(Object ob):** This method appends the specified element to the end of the vector :

**Example :**

```
public class Example1
{
    public static void main(String args[])
    {
        Vector vec = new Vector();
        vec.add("Java");
        vec.add("is");
        vec.add("Fun");
        vec.add(123);
        System.out.println(" Vector is " +vec);
    }
}
```

**Output**

```
Vector is [Java, is, Fun, 123]
[]
```

2. **clear() :** This method is used to remove all the elements of the vector :

**Example:**

```
import java.util.*;

public class Example2
{
    public static void main(String args[])
    {
        Vector vec = new Vector();
        vec.add("Java");
        vec.add("is");
        vec.add("Fun");
        vec.add(123);
        System.out.println(" Vector is " +vec);
        vec.clear();
        System.out.println("After clear :" +vec);
    }
}
```





```
}  
}
```

**Output :**

```
Vector is [Java, is, Fun, 123]  
After clear :[]
```

### 3. Boolean Contains (Object E)

This method return true if the lement is conatined in the vector

```
import java.util.*;  
  
public class Example3  
{  
    public static void main(String args[])  
    {  
        Vector vec = new Vector();  
        vec.add("Java");  
        vec.add("is");  
        vec.add("Fun");  
        vec.add(123);  
        vec.add("Programming");  
        vec.add("Computers");  
  
        if(vec.contains("Fun")==true)  
        {  
            System.out.println("The element is contained in this vector");  
        }  
    }  
}
```

**Output:**

```
The element is contained in this vector
```



#### 4. `indexOf` (Object O):

This method returns the address of the specified element in the vector, or returns -1 if the element is not there :

```
import java.util.*;

public class Example4
{
    public static void main(String args[])
    {
        Vector vec = new Vector();
        vec.add("Java");
        vec.add("is");
        vec.add("Fun");
        vec.add(123);
        vec.add("Programming");
        vec.add("Computers");

        System.out.println("The index of 'Programming' is "
+vec.indexOf("Programming"));
    }
}
```

5. `void insertElementAt(Object E, int index)`: This method is used to insert the element specified into the index specified in the argument :

```
import java.util.*;

public class Example5
{
    public static void main(String args[])
    {
        Vector vec = new Vector();
        vec.add("Java");
        vec.add("is");
        vec.add("Fun");
        vec.add(123);
        vec.add("Programming");
        vec.add("Computers");
    }
}
```



## K. J. Somaiya College of Engineering, Mumbai-77

```
System.out.println("Before inserting: " +vec);
    vec.insertElementAt("Engineering",4);
    System.out.println("After inserting :"+vec);
}
}
```

```
Before inserting: [Java, is, Fun, 123, Programming, Computers]
After inserting :[Java, is, Fun, 123, Engineering, Programming, Computers]
```

### 6. Object lastElement(): Returns the last element of the vector

```
import java.util.*;

public class Example6
{
    public static void main(String args[])
    {
        Vector vec = new Vector();
        vec.add("Java");
        vec.add("is");
        vec.add("Fun");
        vec.add(123);
        vec.add("Programming");
        vec.add("Computers");
        System.out.println("Last Element :"+vec.lastElement());
    }
}
```

```
Last Element :Computers
```

### 7. Void copyInto(Object[] anArray): copies the contents of the invoking vector into the array in the argument.



## K. J. Somaiya College of Engineering, Mumbai-77

```
import java.util.*;

public class Example7
{
    public static void main(String args[])
    {
        Vector vec = new Vector();
        vec.add(1);
        vec.add(2);
        vec.add(3);
        vec.add(4);
        vec.add(5);
        Integer[] arr= new Integer[5];
        vec.copyInto(arr);
        System.out.println("Elements of the array are:");

        for(Integer n : arr)
        {
            System.out.println(n);
        }
    }
}
```

```
Elements of the array are:
1
2
3
4
5
[]
```

- 8. Boolean removeElement(Object ob):** It removes the first occurrence of the object in the vector. Return true if object is found , else returns false.



```
import java.util.*;

public class Example8
{
    public static void main(String args[])
    {
        Vector vec = new Vector();
        vec.add(1);
        vec.add(2);
        vec.add(3);
        vec.add(4);
        vec.add(5);
        System.out.println("Before removing :"+vec);
        vec.removeElement(3);
        System.out.println("After removing :"+vec);
    }
}
```

```
Before removing :[1, 2, 3, 4, 5]
After removing :[1, 2, 4, 5]
❏
```

**9. int size(): This method returns the current size of the vector :**

```
import java.util.*;

public class Example9
{
    public static void main(String args[])
    {
        Vector vec = new Vector();
        vec.add(1);
        vec.add(2);
        vec.add(3);
        vec.add(4);
        vec.add(5);
        System.out.println("size :"+vec.size());
    }
}
```

```
size :5
❏
```



## K. J. Somaiya College of Engineering, Mumbai-77

10.

Boolean addAll(Collection c):

This method adds all the object from the collection to the end of the vector

Returns true if the operation is succesfuul else returns false.

```
import java.util.*;

public class Example10
{
    public static void main(String args[])
    {
        Vector vec = new Vector();
        vec.add(1);
        vec.add(2);
        vec.add(3);
        vec.add(4);
        vec.add(5);
        System.out.println("old vector :"+vec);

        Vector v1 = new Vector();

        v1.add("Blueberry");
        v1.add("Raspberry");
        v1.add("Mango");
        v1.add("Pomegranate");
        v1.add("Apple");

        vec.addAll(v1);

        System.out.println("After Adding:" +vec);
    }
}
```

```
old vector :[1, 2, 3, 4, 5]
After Adding:[1, 2, 3, 4, 5, Blueberry, Raspberry, Mango, Pomegranate, Apple]
[]
```



**K. J. Somaiya College of Engineering, Mumbai-77**



## **K. J. Somaiya College of Engineering, Mumbai-77**