**5 students data should be store in array (ID,Name,Address) and display data value**

class one

{

public static void main(String Args[])

{

Student[] arr;

arr= new Student[5];

arr[0]=new Student(1,"Gautam","Porbandar");

arr[1]=new Student(1,"Yash","Rajkot");

arr[2]=new Student(1,"Saumya","Porbandar");

arr[3]=new Student(1,"Raj","Junagadh");

arr[4]=new Student(1,"Mahesh","Baroda");

for(int i=0;i<5;i++)

{

System.out.println(arr[i].toString());

}

}

}

class Student

{

int id;

String name;

String address;

public Student(int id, String name, String address)

{

this.id=id;

this.name=name;

this.address=address;

}

public String toString()

{

return id + " " + name + " " + address ;

}

}

**sort using bubble sort based based on its employee id**

**Algorithm**

Step 1 : [initialize]

i <- 0

Step 2 : while (i<n-1) repeat thru step 7

Step 3 : j <- 0

Step 4 : While (j<n-i-1) repeat thru step 7

Step 5 : if(emp\_id[j]>emp\_id(j+1))

Then

1. temp <- emp\_id[j]
2. emp\_id[j] <-emp\_id[j+1]
3. emp\_id[j+1]<-temp

Step 6 : j->j+1

Step 7 : i->i+1

Step 8 : Exit

class bubblesort\_empid

{

public static void main(String args[])

{

bubblesort\_empid object = new bubblesort\_empid();

int emp\_id[] = {711,721,529,224,454,987};

object.sortid(emp\_id);

System.out.println("Sorted emp\_iday");

object.sortedid(emp\_id);

}

void sortid(int emp\_id[])

{

int n = emp\_id.length;

for (int i = 0; i < n-1; i++)

for (int j = 0; j < n-i-1; j++)

if (emp\_id[j] > emp\_id[j+1])

{

int temp = emp\_id[j];

emp\_id[j] = emp\_id[j+1];

emp\_id[j+1] = temp;

}

}

void sortedid(int emp\_id[])

{

int n = emp\_id.length;

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

System.out.print(emp\_id[i] + " ");

System.out.println();

}

}

bubbleSort(array)

n = length(array)

repeat

swapped = false

for i = 1 to n - 1

if array[i - 1] > array[i], then

swap(array[i - 1], array[i])

swapped = true

end if

end for

n = n - 1

until not swapped

end bubbleSort

**sort using merge sort based on its employee id**

**Algorithm**

Step 1: Find the middle index of the array.

Middle = 1 + (last – first)/2

Step 2: Divide the array from the middle.

Step 3: Call merge sort for the first half of the array

MergeSort(array, first, middle)

Step 4: Call merge sort for the second half of the array.

MergeSort(array, middle+1, last)

Step 5: Merge the two sorted halves into a single sorted array.

class mergesort\_empid

{

public static void main(String args[])

{

int emp\_id[] = {711,721,529,224,454,987};

mergesort\_empid ob = new mergesort\_empid();

ob.sort(emp\_id, 0, emp\_id.length - 1);

printemp\_iday(emp\_id);

}

void merge(int emp\_id[], int l, int m, int r)

{

int n1 = m - l + 1;

int n2 = r - m;

int L[] = new int[n1];

int R[] = new int[n2];

for (int i = 0; i < n1; ++i)

L[i] = emp\_id[l + i];

for (int j = 0; j < n2; ++j)

R[j] = emp\_id[m + 1 + j];

int i = 0, j = 0;

int k = l;

while (i < n1 && j < n2) {

if (L[i] <= R[j]) {

emp\_id[k] = L[i];

i++;

}

else {

emp\_id[k] = R[j];

j++;

}

k++;

}

while (i < n1) {

emp\_id[k] = L[i];

i++;

k++;

}

while (j < n2) {

emp\_id[k] = R[j];

j++;

k++;

}

}

void sort(int emp\_id[], int l, int r)

{

if (l < r) {

int m =l+ (r-l)/2;

sort(emp\_id, l, m);

sort(emp\_id, m + 1, r);

merge(emp\_id, l, m, r);

}

}

static void printemp\_iday(int emp\_id[])

{

int n = emp\_id.length;

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

System.out.print(emp\_id[i] + " ");

System.out.println();

}

}

**sort using selection sort based on its employee id**

Step 1 : Set Min location at 0

Step 2 : Search the minimum element in the list

Step 3 : Swap with value at location Min

Step 4 : Increment Min to point to next element

Step 5 : Repeat util list is sorted.

class selectionsort\_empid

{

void sort(int emp\_id[])

{

int n = emp\_id.length;

for (int i = 0; i < n-1; i++)

{

int min\_idx = i;

for (int j = i+1; j < n; j++)

if (emp\_id[j] < emp\_id[min\_idx])

min\_idx = j;

int temp = emp\_id[min\_idx];

emp\_id[min\_idx] = emp\_id[i];

emp\_id[i] = temp;

}

}

void printemp\_iday(int emp\_id[])

{

int n = emp\_id.length;

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

System.out.print(emp\_id[i]+" ");

System.out.println();

}

public static void main(String args[])

{

selectionsort\_empid ob = new selectionsort\_empid();

int emp\_id[] = {711,721,529,224,454,987};

ob.sort(emp\_id);

System.out.println("Sorted emp\_iday");

ob.printemp\_iday(emp\_id);

}

}