**Problem Statement:**

1. Create a Student class having roll no. , name, dept., marks. Use array of objects to store details of 5 students. List the name of the student  a) having highest marks b) lowest marks c) Marks more than average.

CODE:

import java.util.\*;

class Student implements Comparable<Student>{

int roll;

String name;

String dept;

double marks;

Student(int roll, String name, String dept, double marks){

this.roll = roll;

this.name = name;

this.dept = dept;

this.marks = marks;

}

@Override

public int compareTo(Student s){

if ( this.marks > s.marks) return 1;

else if (this.marks < s.marks) return -1;

else return 0;

}

}

class Test1{

public static double returnAverage(Student arr[]){

double avg = 0;

for(int i=0; i < arr.length; i++){

avg+= arr[i].marks;

}

return avg/arr.length;

}

public static void main(String[] args) {

Student arr[] = new Student[5];

arr[0] = new Student(3,"Chintu","CSE",85.56);

arr[1] = new Student(5,"Romelu","CSE",85.45);

arr[2] = new Student(7,"Virat","ECE",87.56);

arr[3] = new Student(9,"Lisandro","IT",91.77);

arr[4] = new Student(11,"Raphael","EE",94.14);

System.out.printf("Roll Name Dept. Marks\n");

for (int i=0;i < arr.length ;i++ ) {

System.out.printf("%3d %10s %6s %4f\n",arr[i].roll,arr[i].name,arr[i].dept,arr[i].marks);

}

Arrays.sort(arr);

System.out.println("Highest marks is " + arr[4].marks + " received by " + arr[4].name );

System.out.println("Lowest marks is " + arr[0].marks + " received by " + arr[0].name );

double avg = returnAverage(arr);

System.out.println("Average marks is : " + avg);

System.out.println("Marks greater than average : ");

for (int i=0;i < arr.length ;i++ ) {

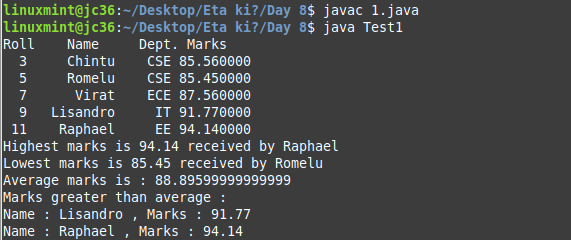
if (arr[i].marks > avg) System.out.println("Name : " + arr[i].name + " , Marks : " + arr[i].marks);

}

}

}

OUTPUT:



2.Implement 2 sorting algorithms for sorting student array.

CODE:

import java.util.\*;

class Student{

int roll;

String name;

String dept;

double marks;

Student(int roll, String name, String dept, double marks){

this.roll = roll;

this.name = name;

this.dept = dept;

this.marks = marks;

}

}

class Test2{

public static void bubbleSort(Student arr[]){

for(int i=0; i < arr.length - 1; i++){

for(int j=0; j < arr.length - i - 1; j++){

if (arr[j].marks > arr[j+1].marks){

Student temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

public static void insertionSort(Student arr[]){

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

Student key = arr[i];

int j = i-1;

while (j >=0 && arr[j].roll > key.roll) {

arr[j+1] = arr[j];

j = j - 1;

} arr[j+ 1] = key;

}

}

static void swap(Student arr[], int i, int j){

Student temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

static int partition(Student[] arr, int low, int high){

double pivot = arr[high].marks;

int i = (low - 1);

for(int j = low; j <= high - 1; j++){

if (arr[j].marks > pivot){

i++;

swap(arr, i, j);

}

}

swap(arr, i + 1, high);

return (i + 1);

}

static void quickSort(Student arr[], int low, int high){

if (low < high){

int pi = partition(arr, low, high);

quickSort(arr, low, pi - 1);

quickSort(arr, pi + 1, high);

}

}

public static void main(String[] args) {

Student arr[] = new Student[5];

arr[0] = new Student(3,"Chintu","CSE",85.56);

arr[1] = new Student(5,"Romelu","CSE",85.45);

arr[2] = new Student(7,"Virat","ECE",87.56);

arr[3] = new Student(9,"Lisandro","IT",91.77);

arr[4] = new Student(11,"Raphael","EE",94.14);

System.out.printf("Roll Name Dept. Marks\n");

for (int i=0;i < arr.length ;i++ ) {

System.out.printf("%3d %10s %6s %4f\n",arr[i].roll,arr[i].name,arr[i].dept,arr[i].marks);

}

bubbleSort(arr);

System.out.println("Sorted array (according to marks) after applying Bubble Sort :");

System.out.printf("Roll Name Dept. Marks\n");

for (int i=0;i < arr.length ;i++ ) {

System.out.printf("%3d %10s %6s %4f\n",arr[i].roll,arr[i].name,arr[i].dept,arr[i].marks);

}

insertionSort(arr);

System.out.println("Sorted array (according to roll) after applying Insertion Sort :");

System.out.printf("Roll Name Dept. Marks\n");

for (int i=0;i < arr.length ;i++ ) {

System.out.printf("%3d %10s %6s %4f\n",arr[i].roll,arr[i].name,arr[i].dept,arr[i].marks);

}

System.out.println("Reverse sorted array (according to marks) after applying Quick Sort :");

quickSort(arr,0, arr.length - 1);

System.out.printf("Roll Name Dept. Marks\n");

for (int i=0;i < arr.length ;i++ ) {

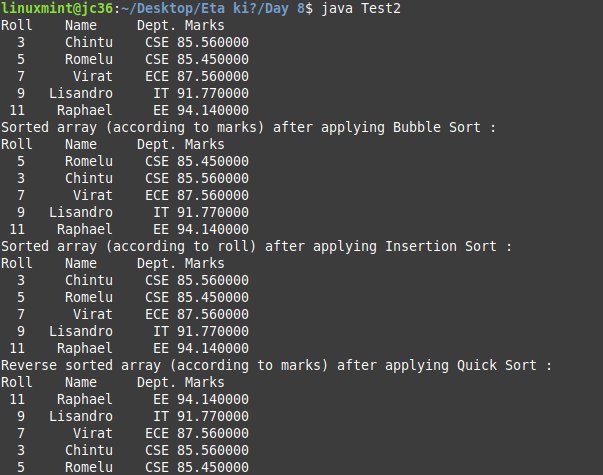
System.out.printf("%3d %10s %6s %4f\n",arr[i].roll,arr[i].name,arr[i].dept,arr[i].marks);

}

}

}

OUTPUT:



3. Use wrapper class for explaining auto-boxing and unboxing.

CODE :

import java.util.\*;

class Test3{

public static void main(String[] args) {

Character a = 'f';

// unboxing in java

char unbox = a;

System.out.println("Unboxing in Java : ");

System.out.println("Character a = " + a);

System.out.println("char unbox = " + unbox);

// autoboxing in java

int n = 9;

Integer na = n;

System.out.println("Autoboxing in Java : ");

System.out.println("int n = " + n);

System.out.println("Integer na = " + n);

// autoboxing in ArrayList

ArrayList<Integer> al = new ArrayList<Integer>();

al.add(1);

al.add(450);

al.add(69);

System.out.println(al);

}

}OUTPUT:

