

Question 1: (Array of objects)

10 Marks

Problem description: A coffee shop would like to find out the customer feedback rating about its services. The customer class shown below:

Customer
-name: String -feedbackRating:double +Customer() +Customer(String name, double feedbackRating) +getters() +setters()

Example: Assume that the shop will collect feedback from 'N' customers. Following are the sample customer feedback values.

Customer 1 - 3 out of 5

Customer 2 - 4 out of 5

Customer 3 - 2.5 out of 5

Write a method which accepts array of 'N' customer objects to store feedback values of these customers and return average feedback rating.

Input

Input array of 'N' customer objects

Output

Output is average feedback value.

Test Case ID	Input	Output	Marks
UTC01_01	Customer1 {Vijay, 3.0} Customer2 {Kumar, 4.0} Customer3 {Vishnu, 5.0}	4.0	10
UTC01_02	Customer1 {Arun, 4.5} Customer2 {Babu, 4.5} Customer3 {Manasa, 3.5} Customer 4 {Sara, 2.0} Customer 5 {Kumar, 4.0}	3.7	
UTC01_03	Customer1 {Santosh, 5.0} Customer2 {Savitha, 0.0}	2.5	

Question2 (Comparison of objects)

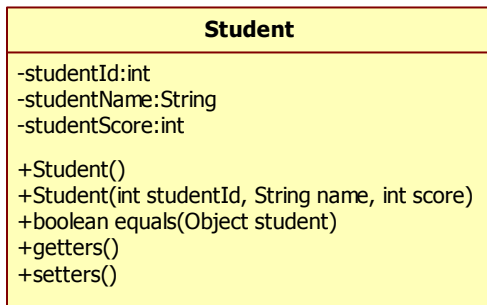
10 Marks

Problem description: A college admin would like to compare students of two batches based on their semester total score. Implement following method:

- boolean compareStudents(Student batch1[], Student batch2[]); This will return “true” when both array of objects are equal based on score value otherwise it returns “false”

Below is the Student class diagram.

—

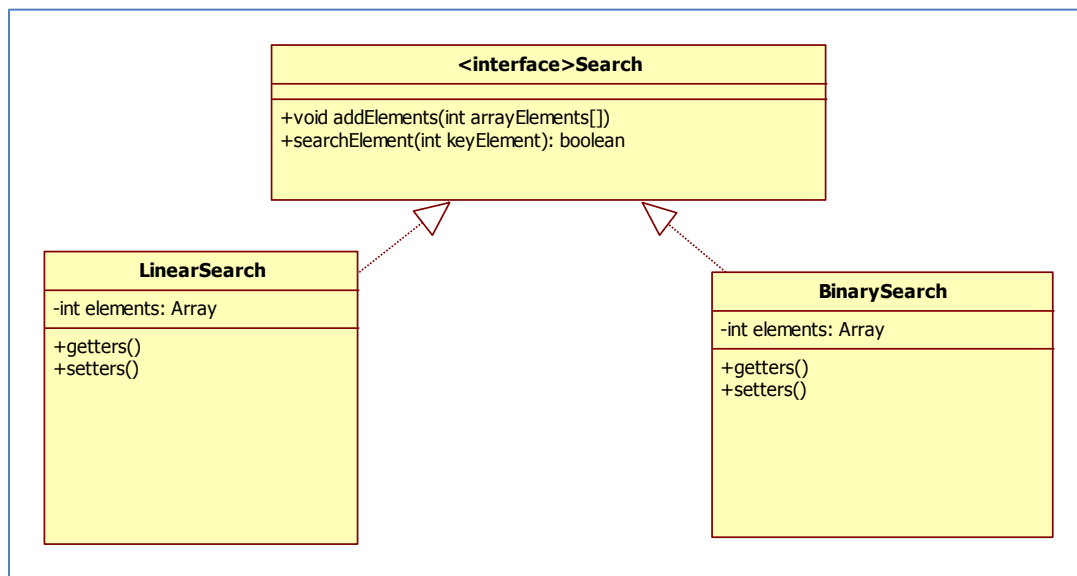


Test Case ID	Input	Output	Marks
UTC02_01	batch1 { (100,"kumar",56), (200,"Santosh",60),(300,"Sharat",70)} batch2 { (500,"Dhoni",56), (501,"Raina",60),(502,"Vijay",70)}	true	10
UTC02_02	batch1 { (600,"Kiran",90), (700,"Kushal",95),(800,"Senthil",90)} batch2 { (701,"Anjali",90), (702,"Anu",95),(703,"Abida",90)}	true	
UTC02_03	batch1 { (600,"Vikram",70), (700,"Manas",85),(800,"Girish",90)} batch2 { (701,"Ranjan",70), (702,"Sharat",65)}	false	

Question3 (Interface)

10 Marks

Problem description: Write a program which implements following UML class diagram:



1. `addElements()` – Should implement adding elements to array 'elements'
2. `searchElement(int key)` – Should search array for given 'key' element. If found then it must return 'true' otherwise return 'false'.

Implement following method in class called "SearchElements" to test above code by creating appropriate object.

- `boolean searchGivenElements(Search search, int key);`

Here 'search' object could of type 'LinearSearch' or 'BinarySearch' and 'key' is the element to be searched.

Test Case ID	Input	Output	Marks
UTC03_01	Array Elements={10,15,5,20} Key = 15 Search object = 'LinearSearch'	true	10
UTC03_02	Array Elements={100,150,500,720,800,900} Key = 1300 Search object = 'BinarySearch'	false	
UTC03_03	Array Elements={45,67,89,90,110,123,456} Key = 456 Search type = ' BinarySearch '	true	