

# JAVA Test

1. We have the scores of the last 7 cricket matches India has played. For the purpose of analyzing their performance, we need to find the average score, the number of scores below it, those equal to it, and those above it.

Consider that the scores (runs) are stored in an array of int values.

Create a class CricketScore and write a program to implement the above requirement.

Sample:

Input:

```
int score = {281, 344, 265, 272, 236, 324, 287};
```

Output:

The average score of the team is 287 runs

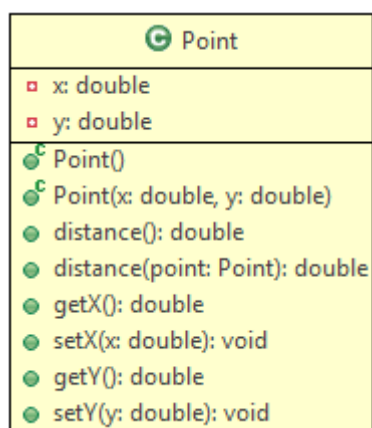
No. of matches having a score above average is 2

No. of matches having a score equal to average is 1

No. of matches having score below average is 4

2. The Point class represents a point with x and y coordinates. Create a class Point according to the class diagram given below:

**Point:**



**Method Description:**

distance(): This method returns the distance of this point from origin.

distance(Point point): This method returns the distance of this point from the specified point.

Use the formula given below to implement the methods:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Hint: Use Math.sqrt(double d) method to calculate the square root, and Math.round(double d) method to round off the values.

To do: Make a Tester class and test the following functionalities:

- Create an object p1 of Point class
- Set the values of x and y
- display the distance of the point from origin
- Create another object p2 of Point class
- Set the values of x and y
- Display the distance between p1 and p2

Note: You can use the Math class for calculations.

Sample:

Input(for p1): x=2,y=3

Input(for p2): x=5,y=6

Output:

The distance of p1 from p2 is 4.242640687119285  
The distance of p1 from the origin is 3.605551275463989

3. Create a class LuckyNumber which checks if the input is a Lucky number. A lucky number is a number where the sum of squares of every even-positioned digit (starting from the second position) is a multiple of 9.

**For e.g.  $1623 = 6^2 + 3^2 = 45$  is a multiple of 9 and hence is a Lucky number**

Implement the above program using appropriate iterative and conditional control structures.

**Sample:**

**Input:** 1623

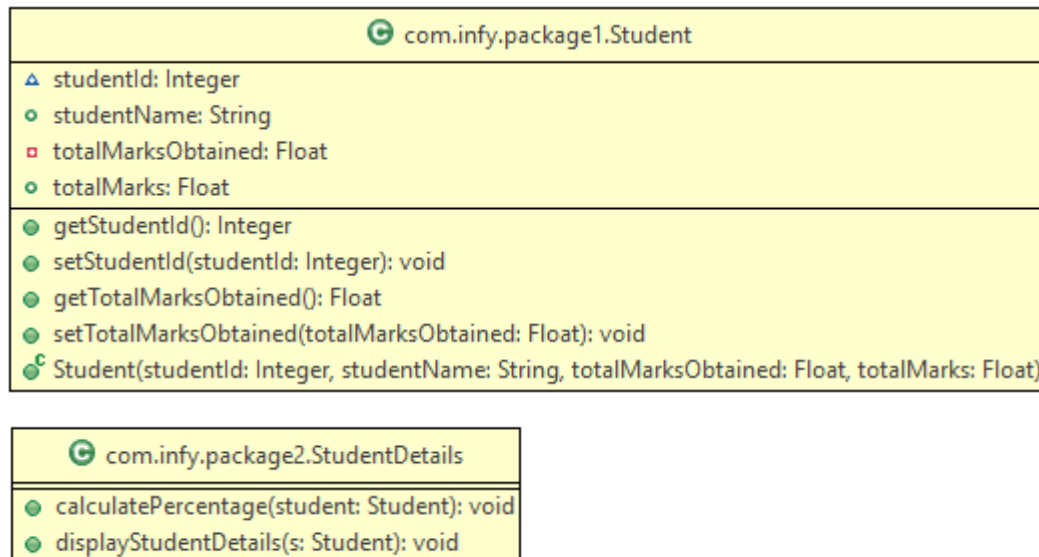
**Output:** The number 1623 is a Lucky number

**Input:** 1523

**Output:** Oops! Not a Lucky number

4. Create a new Java Project with one package named **com.infy.package1** which contains the **Student** Class and another package named **com.infy.package2** which contains the **StudentDetails** class according to the class diagrams given below.

The Student class contains a parametrized constructor to initialize its instance variables. It also contains getters and setters for its non-public attributes.

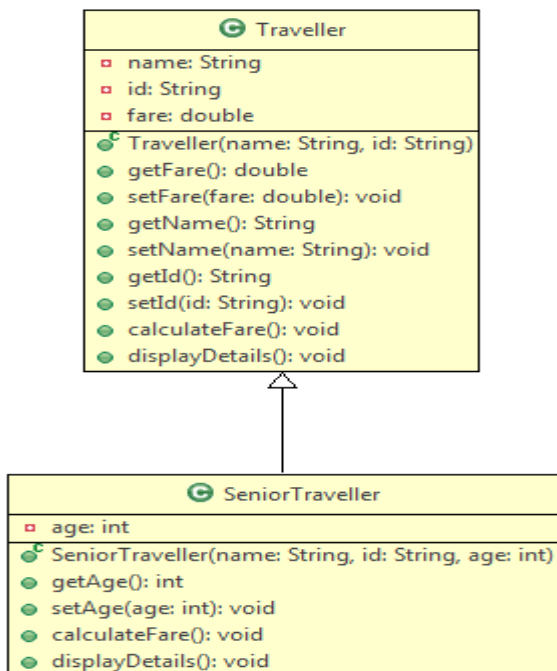


The StudentDetails class contains 2 methods i.e. **calculatePercentage()** and **displayStudentDetails()**. The **calculatePercentage()** calculates and prints the percentage of the student whose details are passed to it. The **displayStudentDetails()** prints all the student details including the **studentId**, **studentName**, **totalMarks** and **totalMarksObtained** in the format given in the expected result.

**com.infy.package2** contains a class called **Tester** which creates a new Student object. It calls the **displayStudentDetails** to print the student details and then calls the **calculatePercentage()** to print the calculated percentage of the student object in the proper format according to the expected result.

5. Soudagar Travels is one of the most popular travel agencies in Karnataka. It has launched a limited period tour offer on a specific travel route - Mysore to Munnar, for which the base fare has been fixed at Rs. 2000. Also, a service tax of 11.36% is charged on the final ticket amount.

Id proof is required for the travel, and it can be either Voter Id, PAN or Aadhaar number. Create the classes to represent travellers and senior travellers according to the class diagrams below:



## Method Description:

### Traveller:

- **calculateFare():** Calculates and sets the fare by adding the service tax to the base fare
- **displayDetails():** Displays the details of the traveller as shown in the sample output.
  - Use calculateFare() method to calculate the fare

### SeniorTraveller:

- calculateFare(): Calculates and sets the fare for senior travellers. Senior travellers are eligible for a discount according to the following table:

Age	Discount(%)
50 to 65 (both inclusive)	10
Greater than 65	15

- Provide discount on the base fare according to the above table, and add service tax on the discounted fare
- If the age is below 50, set the fare as -1
- Use @Override on this method
- **displayDetails():** Displays the details of the senior traveller as shown in the sample output.
  - Use calculateFare() method to calculate the fare

- If someone signs up for being a senior citizen traveller but is below 50 years of age, an appropriate error message has to be shown
- Use @Override on this method

Create a Tester class to test the above functionalities. Create Traveller and SeniorTraveller objects with values as given below, and use the displayDetails() method to show the result.

### Sample Input:

#### Traveller

Attribute	Value
Name	Roger
ID	AQW1344321

#### Senior Traveller

Attribute	Value
Name	Joey
Id	PJAMG7755TY
Age	70

### Sample Output:

Traveller Details

\*\*\*\*\*

Traveller name : Priya

Traveller Id : AQW1344321

Cost of Travel : 2227.2

Senior Traveller Details

\*\*\*\*\*

Traveller name : Joey

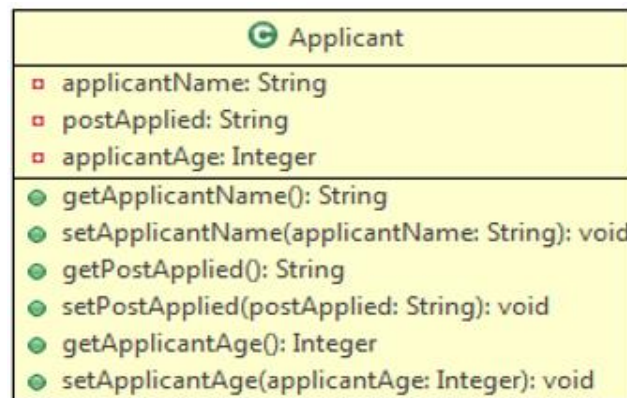
Traveller Id : PJAMG7755TY

Cost of Travel : 1893.12

6. Infy Bank wants to conduct examinations for the post of Probationary Officers, Assistants, and Special Cadre Officers. It has rolled out an online application which is available on the

Bank's website. The applicants can fill in the application form and submit it with accurate details.

Assuming that each **Applicant** is represented by the following class diagram:



Define an user-defined Exception **InfyBankException** according to following class

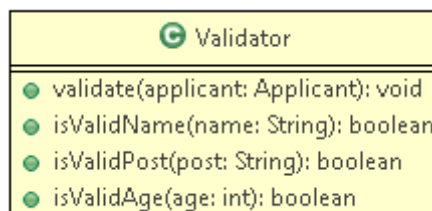


diagram:

Design a **Validator** class which has methods for validating applicant details according to the following class diagram :

Violation for	User defined exception	Exception message
Applicant name	InvalidNameException	Invalid applicant name
Post	InvalidPostException	Invalid post
Age	InvalidAgeException	Invalid age exception

### Method Description:

- **validate(Applicant applicant):**
  - This method receives the Applicant and calls the respective methods to validate the values. If validation fails, it throws user-defined exceptions **InfyBankException** with the exception message as given below:

If all values are valid print the following message:

All the values are valid.

### **isValidApplicantName(String name):**

- This method validates applicantName.
- It cannot be null or empty.

- If the rule is violated then it should return false else it should return true.
- **isValidPost(String post):**
  - This method validates the post the applicant applied for.
  - It should be one among one of the following posts: "Probationary Officer", "Assistant", or "Special Cadre Officer".
  - If the rule is violated then it should return false else it should return true.
- **isValidAge(Integer age):**
  - This method validates the age of the applicant.
  - It should be greater than 18 years and less than 35 years.
  - If the rule is violated then it should return false else it should return true.

Create a class **Tester** and implement it as shown below:

1. Create an object of the Applicant class and populate it with values.
2. Invoke validate() method of Validator class to validate the values.
3. If any exception is thrown, catch the exception and print the exception message.

Note: You can change the input as needed to get a different output.

**Sample:**

**Input:**

<b>Name</b>	Jason
<b>Post</b>	Assistant
<b>Age</b>	37

**Output:** Invalid age exception