













**BIKE SHOWROOM - REQUIREMENT 1**

**Bike Showroom Requirement - 1**

**Requirement 1:**

Let’s start off by creating two **Bike** objects and check whether they are equal.

Create a **Bike** Class with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| VIN | String |
| brand | String |
| model | String |
| engineDisplacement | String |
| brakeSystem | String |
| cost | Double |

Mark all the attributes as private

Create / Generate appropriate Getters & Setters

Add a parameterized constructor to take in all attributes in the given order:   
**Bike(String VIN, String brand, String model,String engineDisplacement, String brakeSystem, Double cost)**

When the “Bike” object is printed, it should display the following details: **[Override the toString method]**  
Print format:  
VIN:"VIN"  
Brand:"brand"  
Model:"model"  
Engine Displacement:"engineDisplacement"  
Brake System:"brakeSystem"  
Cost:"cost"

Two Bikes are considered same if they have the same VIN  and brand(Case – Insensitive). Implement the logic in the appropriate function. **[Override the equals method]**  
  
The input format consists of Bike details separated by comma in the below order,  
**VIN,brand,model,engineDisplacement,brakeSystem,cost**

The Input to your program would be details of two Bike, you need to display their details as given in "5th point(refer above)" and compare the two Bikes and display if the Bikes are same or different.  
  
**Note:**There is an empty line between display statements. Print the empty lines in main function.  
             Display single decimal value for double datatype  
  
**Sample Input & Output 1:**  
  
Enter bike 1 details:  
**MD2GJ3214JR258416,KTM,RC,390cc,Disk,225000.0**  
Enter bike 2 details:  
**MD2GJ3214JR258416,KTM,RC,390cc,Disk,225000.0**  
  
Bike 1  
VIN:MD2GJ3214JR258416  
Brand:KTM  
Model:RC  
Engine Displacement:390cc  
Brake System:Disk  
Cost:225000.0  
  
Bike 2  
VIN:MD2GJ3214JR258416  
Brand:KTM  
Model:RC  
Engine Displacement:390cc  
Brake System:Disk  
Cost:225000.0  
  
Bike 1 is same as Bike 2  
  
**Sample Input & Output 2:**  
  
Enter bike 1 details:  
**MD2GJ3214JR258416,KTM,RC,390cc,Disk,225000.0**  
Enter bike 2 details:  
**WS22UA847JK874510,BMW,S,1000cc,Disk,1800000.0**  
  
Bike 1  
VIN:MD2GJ3214JR258416  
Brand:KTM  
Model:RC  
Engine Displacement:390cc  
Brake System:Disk  
Cost:225000.0  
  
Bike 2  
VIN:WS22UA847JK874510  
Brand:BMW  
Model:S  
Engine Displacement:1000cc  
Brake System:Disk  
Cost:1800000.0  
  
Bike 1 and Bike 2 are different

**BIKE SHOWROOM - REQUIREMENT 2**

**Bike Showroom Requirement - 2**

**Requirement 2:**  
In this requirement, you need to validate the VIN of the bike.  
  
a)Create a class **Main** with the following static methods:

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| static Boolean validateVIN(String vin) | Validate the VIN based on the rules given below. Returns **true**ifVIN is valid else return **false** |

b) While validating VIN(Vechile Identification Number) follow the below rules. The format of the VIN is given below  
  
Sample format**:1HGBH4G24MN109186**  
  
1.Length of the VIN should seventeen  
2. The first three ,seventh ,tenth and eleventh characters could contains either alphabets(uppercase) or  numeric values.  
3. The fourth and fifth should contains only alphabets(uppercase).  
4. Rest of the characters should contains only numeric values.  
  
**Note:** Print "**VIN is valid**" if refId is valid else print **"VIN is invalid"**.  
            All the above print statements are present in the main method.  
  
**[All text in bold corresponds to input]**  
**Sample Input and Output 1:**  
  
Enter the VIN to be validated:  
**ME1CF2384HN874621**  
VIN is valid  
  
**Sample Input and Output 2:**  
  
Enter the VIN to be validated:  
**MD24H4341JD471245**  
VIN is invalid

Select a Language to Start Solving



[GET EDITOR](javascript:void(0))

**BIKE SHOWROOM - REQUIREMENT 3**

**Bike Showroom - Requirement 3**

**Requirement 3:**  
  
In this requirement, you should find the city wise count of the showrooms.  
  
a) Create a Class **Showroom** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| id | String |
| name | String |
| email | String |
| brand | String |
| city | String |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a parameterized constructor to take in all attributes in the given order:**Showroom(String id, String name, String email, String brand, String city)**  
  
b) Create the following static methods in the **Showroom**class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| static Map<String,Integer> cityWiseShowroomCount(List<Showroom> list) | This method accepts a List of Showroom objects. It returns a Map with city name as key and the number of showrooms in the city as value in alphabetical order of city name. |

The showroom details should be given as a comma-separated value in the below order,  
**id, name, email, brand, city (the id of the showroom contains '#')**  
  
**Note:**  
Use **System.out.format("%-15s %s\n","City","Count")** for formatting the output.  
Use **TreeMap**in the **cityWiseShowroomCount**method.  
  
**Sample Input/Output 1:**  
  
Enter the number of showrooms  
**5  
#8741BLR,Jack Suzuki,jacksonsuzuki@yahoo.com,Suzuki,Bangalore  
#4785BLR,Balaji Bajaj,balaji.bajaj@yahhoo.com,Bajaj,Bangalore  
#5478HSRA,Sai Ram Bajaj,srirambajaj@gmail.com,Bajaj,Hosur  
#4123CHN,Kick start Suzuki,kickstartsuzuki@gmail.com,Suzuki,Chennai  
#6541CHN,Benjamin Honda,benjaminhonda@gmail.com,Honda,Chennai**  
City                  Count  
Bangalore       2  
Chennai           2  
Hosur               1

**BIKE SHOWROOM - REQUIREMENT 4**

**Bike Showroom - Requirement 4**

**Requirement 4:**  
  
In this requirement, you need to sort the list of bikes based on brand, and engineDisplacement.  
  
a) Create a Class **Bike** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| VIN | String |
| brand | String |
| model | String |
| engineDisplacement | String |
| brakeSystem | String |
| cost | Double |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a parameterized constructor to take in all attributes in the given order: **Bike( String VIN, String brand, String model, String engineDisplacement, String brakeSystem, Double cost )**  
  
b) Create the following static methods in the Bike class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| static Bike createBike(String detail) | This method accepts a String. The bike detail separated by commas is passed as the argument. Split the details and create a bike object and returns it. |

The bike details should be given as a comma-separated value in the below order,  
**VIN, brand, model, engineDisplacement, brakeSystem, cost**  
  
c) The Bike class should implement the **Comparable** interface which sorts the Bike list based on brand. While comparing, all the brand attributes in the list are unique.  
  
d) Create a class **EngineDisplacementComparator** which implements Comparator interface and sort the Bike list based on engineDisplacement. While comparing, all the engineDisplacement attributes in the list are unique.  
  
Get the number of Bikes and bike details and create a bike list. Sort the Bikes according to the given option and display the list.  
  
When the “bike” object is printed, it should display the following details  
Print format:  
**System.out.format("%-20s %-10s %-10s %-20s %-12s %s\n","VIN","Brand","Model","Engine Displacement","Brake System","Cost");**  
Display one digit after decimal point for Double datatype.  
  
**Sample Input and Output 1:**

Enter the number of the bikes:  
**3  
MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0  
ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0  
MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
Enter a type to sort:  
1.Sort by Brand  
2.Sort by Engine Displacement  
**1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VIN | Brand | Model | Engine Displacement | Brake System | Cost |
| MD2NH43DLJD471245 | Bajaj | Pulsar | 150cc | Disk | 88000.0 |
| MD6NF25FXHA546812 | TVS | Star City | 110cc | Drum | 49000.0 |
| ME1CF23XBHN874621 | Yamaha | Ray | 120cc | Drum | 55000.0 |

**Sample Input and Output 2:**

Enter the number of the bikes:  
**3  
MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0  
ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0  
MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
Enter a type to sort:  
1.Sort by Brand  
2.Sort by Engine Displacement  
**2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VIN | Brand | Model | Engine Displacement | Brake System | Cost |
| MD6NF25FXHA546812 | TVS | Star City | 110cc | Drum | 49000.0 |
| ME1CF23XBHN874621 | Yamaha | Ray | 120cc | Drum | 55000.0 |
| MD2NH43DLJD471245 | Bajaj | Pulsar | 150cc | Disk | 88000.0 |

**BIKE SHOWROOM - REQUIREMENT 5**

**Bike Showroom - Requirement 5**

**Requirement 5:**  
In this requirement develop a feature in which you can search a List of Bikes by engineDisplacement, and cost.  
  
a) Create a Class Bike with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| VIN | String |
| brand | String |
| model | String |
| engineDisplacement | String |
| brakeSystem | String |
| cost | Double |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a parameterized constructor to take in all attributes in the given order: Bike( String VIN, String brand, String model, String engineDisplacement, String brakeSystem, Double cost )  
  
b) Create a class **BikeBO**with the following methods,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public List<Bike> findBike(List<Bike> bikeList,String engineDisplacement) | This method accepts a list of bikes and engineDisplacement as arguments and returns a list of bikes that matches with the given engineDisplacement. |
| public List<Bike> findBike(List<Bike> bikeList,Double cost) | This method accepts a list of bikes and cost as arguments and returns a list of bikes that has cost less than the given cost. |

The bike details should be given as a comma-separated value in the below order,  
VIN, brand, model, engineDisplacement, brakeSystem, cost  
  
Print format:  
**System.out.format("%-20s %-10s %-10s %-20s %-12s %s\n","VIN","Brand","Model","Engine Displacement","Brake System","Cost");**  
  
**Note:**The bike lists are displayed in the main method.  
            If any other choice is selected, display "**Invalid choice**"  
            If the search detail is not found, display "No such bike is present"  
            Display one digit after the decimal point for Double Datatype.

**Sample Input/Output 1:**  
   
Enter the number of bikes:  
**5**  
**MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0**  
**ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0**  
**MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
**MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0**  
**WS22UA84NJK874510,BMW,S,1000cc,Disk,1800000.0**  
Enter a search type:  
1.By Engine Displacement  
2.By Cost  
**1**  
Enter the engine displacement:  
**150cc**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VIN | Brand | Model | Engine Displacement | Brake System | Cost |
| MD2NH43DLJD471245 | Bajaj | Pulsar | 150cc | Disk | 88000.0 |
| MA4NH43BNJV412631 | Suzuki | Gixxer | 150cc | Disk | 90000.0 |

**Sample Input/Output 2:**  
   
Enter the number of bikes:  
**5**  
**MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0**  
**ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0**  
**MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
**MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0**  
**WS22UA84NJK874510,BMW,S,1000cc,Disk,1800000.0**  
Enter a search type:  
1.By Engine Displacement  
2.By Cost  
**2**  
Enter the cost:  
**50000**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VIN | Brand | Model | Engine Displacement | Brake System | Cost |
| MD6NF25FXHA546812 | TVS | Star City | 110cc | Drum | 49000.0 |

   
**Sample Input/Output 3:**  
   
Enter the number of bikes:  
**5**  
**MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0**  
**ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0**  
**MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
**MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0**  
**WS22UA84NJK874510,BMW,S,1000cc,Disk,1800000.0**  
Enter a search type:  
1.By Engine Displacement  
2.By Cost  
**1**  
Enter the engine displacement:  
**200cc**  
No such bike is present  
   
**Sample Input/Output 4:**  
   
Enter the number of bikes:  
**5**  
**MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0**  
**ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0**  
**MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
**MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0**  
**WS22UA84NJK874510,BMW,S,1000cc,Disk,1800000.0**  
Enter a search type:  
1.By Engine Displacement  
2.By Cost  
**4**  
Invalid choice

**BIKE SHOWROOM - REQUIREMENT 6**

**Bike Showroom - Requirement 6**

**Requirement 6:**  
In this requirement develop a feature in which you can find the best showroom among the given list of showroms.  
  
a) Create a class **Showroom** with the following attributes,

|  |  |
| --- | --- |
| **Member Field Name** | **Data Type** |
| id | String |
| name | String |
| email | String |
| brand | String |
| city | String |
| bikes | List<Bike> |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a parameterized constructor to take in all attributes in the given order: **Showroom(String id, String name, String email, String brand, String city, List<Bike> bikes)**  
  
b) Create a class **Bike** with the following attributes,

|  |  |
| --- | --- |
| **Member Field Name** | **Data Type** |
| VIN | String |
| brand | String |
| model | String |
| engineDisplacement | String |
| brakeSystem | String |
| cost | Double |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a parameterized constructor to take in all attributes in the given order: **Bike( String VIN, String brand, String model, String engineDisplacement, String brakeSystem, Double cost )**  
  
c) Create the following static method in the Showroom class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| Showroom bestShowroom(List<Showroom> showroomList) | this method takes showroom details as argument and returns the best showroom based on the below rules. |

d) The showroom with wide variety of bikes ie., unique bikes with different specifications (engineDisplacement) is said to be the best showroom.  
  
For Example, consider a showroom A which has 4 different bikes with 120cc, 150cc, 150cc, 150cc and showroom B has 3 different bikes with 150cc, 180cc, 210cc, then showroom B is said to be best showroom as it has more number of unique variety of bikes for the customer to select.  
  
The showroom details should be given as a comma-separated value in the below order,  
id,name,email,brand,city (the id of the showroom contains '#')  
  
The bike details should be given as a comma-separated value in the below order,  
VIN, brand, model, engineDisplacement, brakeSystem, cost  
  
**Note:**All print lines are displayed in the main method.  
  
**Sample Input/Output :**  
  
Enter the number of showroom:  
**4**  
Enter the showroom details:  
**#1010BLR,Aditya Honda,adityahonda@gmail.com,Honda,Bangalore**  
Enter the number of bikes:  
**3  
MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0  
ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0  
MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
Enter the showroom details:  
**#1102HSRA,Sundhram TVS,sundhramtvs@yahoo.com,TVS,Hosur**  
Enter the number of bikes:  
**4  
MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0  
MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0  
ME4NH42ANJR985641,Honda,CBR,150cc,Disk,125000.0  
MD2GJ32NBJR258416,KTM,RC,390cc,Disk,225000.0**  
Enter the showroom details:  
**#1546CHN,Racer KTM,racer.ktmchennai@yahoo.com,KTM,Chennai**  
Enter the number of bikes:  
**3  
MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0  
ME4NH42ANJR985641,Honda,CBR,150cc,Disk,125000.0  
MD2GJ32NBJR258416,KTM,RC,390cc,Disk,225000.0**  
Enter the showroom details:  
**#1452MB,PU Bajaj,pu.bajaj@gmail.com,Bajaj,Mumbai**  
Enter the number of bikes:  
**4  
ME4NH42ANJR985641,Honda,CBR,150cc,Disk,125000.0  
MD2GJ32NBJR258416,KTM,RC,390cc,Disk,225000.0  
JKA1KA12NJA645125,Ninja,RC,300cc,Disk,250000.0  
WS22UA84NJK874510,BMW,S,1000cc,Disk,1800000.0**  
The best Showroom is PU Bajaj