

Task 1

Create a java class file called Student.java. The Student.java class should contain the following members:

1. a private field to store the class roll number of a student, as a primitive byte type value.
2. a private field to store the first name of a student, as a reference String type value. **This should be immutable!**
3. a private field to store the last name of a student, as a reference String type value. **This should be immutable!**
4. a private field to store the current year of a student, as a primitive byte type value.
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called PortalCredentialsCreator.java. The class should contain the following two methods:

1. `public List<UserName> createListOfUserNames(List<Student> studentsList) {}`

This method inputs a list of Student type defined above, and creates a list of UserName type. Create a java class file called UserName.java and create a private field called userName, to store the user name of a student as a reference String value. No two user names in the list should be the same.

2. `public List<PassWord> createListOfRandomPasswords(List<Student> studentsList) {}`

This method inputs a list of Student type defined above, and creates a list of PassWord type. Create a java class file called PassWord.java and create a private field called passWord, to store the password of a student as a reference String value. The value of a password should be a random string of length 8-20.

Create a separate execution java class called Main.java and call both the methods given above!

Task 2

Create a java class file called Student.java. The Student.java class should contain the following members:

1. a private field to store the class roll number of a student, as a primitive byte type value.
2. a private field to store the name of a student, as a reference String type value. **This should be immutable!**
3. a private field to store the parent's/guardian's contact number of a student, as a primitive long type value. **This should be immutable!**
4. a private field to store the marks, as a primitive double type value.
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called TestResultGenerator.java. The class should contain the following two methods:

1. `public HashSet<String> getDetailsOfFailedStudents(Set<Student> studentSet) {}`

This method inputs a Set of Student type and returns a HashSet of String type. A single element in the returning HashSet will be a String that contains the concatenated name and parent's contact number of a student, e.g.

```
["Divyansh-1234657890", "Vaibhav-1118529630", "Neeraj-1593574568"]
```

2. `public Set<Student> sortStudentsForRanking(HashSet<Student> studentSet) {}`

This method inputs a Set of Student type and returns a Set of Student type. This method is supposed to sort the HashSet of students in an increasing manner of their marks. Since the marks are primitive double type, you are supposed to use their natural ordering. If two students have same marks, use class roll number.

Create a separate execution java class called Main.java and call both the methods given above!

Task 3

Create a java class file called Employee.java. The Employee.java class should contain the following members:

1. a private field to store the employee id of an employee, as a primitive short type value. **This should be immutable!**
2. a private field to store the name of an employee, as a reference String type value. **This should be immutable!**
3. a private field to store the salary of an employee, as a primitive double type value.
4. a private field to store the number of leaves taken by an employee, as a primitive byte type value.
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called EmployeeLeaveCalculator.java. The class should contain the following two methods:

1. `public List<Short> getDefaultlerEmployeeIDList(List<Employee> employeeList) {}`

This method inputs a list of Employee as an argument and returns a list of employee IDs (as short) where the employee is a "defaulter", an employee is considered to be a defaulter, if they have taken more than 25 leaves. Make sure that the list does not contain duplicate elements! Decrease their salary by 1%.

2. `public ArrayList<String> getNumberOfLeaves(List<Employee> employeeList) {}`

This method inputs a list of Employee type as an argument and returns an ArrayList of String type. The String in the ArrayList will contain the name of the employee and the total number of leaves taken by the employee.

e.g ["Divyansh-5", "Neeraj-5", "Vaibhav-8"] Sort this list according to the number of leaves, use name for ambiguity.

Create a separate execution java class called Main.java and call both the methods given above!

Task 4

Create a java class file called `YouTubeVideo.java`. The `YouTubeVideo.java` class should contain the following members:

1. a private field to store the video id of the video, as a primitive long type value. **This should be immutable!**
2. a private field to store the name of the video, as a reference String type value. **This should be immutable!**
3. a private field to store the number of likes, as a reference BigInteger type value.
4. a private field to store the duration of video in minutes, as a primitive int type value.
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the `equals()`, `toString()` and the `hashCode()` methods.

Create a java class file called `VideoAnalytics.java`. The class should contain the following two methods:

1. `public HashSet<YouTubeVideo> getTopTenViralVideos(List<YouTubeVideo> videos) {}`

This method inputs a list of `YouTubeVideo` type and returns a hash set of `YouTubeVideo` type. This method will iterate through the list of videos it has received and it should find out the top 10 videos with the most number of likes, then store those videos in a `HashSet` and return it. If two videos have the same number of likes, use duration.

2. `public List<String> showDetailsOfShortVideos(Set<YouTubeVideo> videos) {}`

This method inputs a set of `YouTubeVideo` type and returns a list of String type. A short video is one whose duration is less than or equal to 10 minutes. Find all such videos and store the name, number of likes and duration of the video into a String, and store all such strings into a list and return it from the method.

Ex = ["Java for Beginners - 10000 - 10", ...]

Create a separate execution java class called `Main.java` and call both the methods given above!

Task 5

Create a java class file called Teacher.java. The Teacher.java class should contain the following members:

1. a private field to store the employee ID of a teacher, as a reference String type value. **IMMUTABLE!**
2. a private field to store the name of a teacher, as a reference String type value. **IMMUTABLE!**
3. a private field to store the salary of a teacher, as a primitive double type value.
4. a private field to store the list of subjects that the teacher teaches, as a List<String> (list of String).
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called TeacherHelper.java. The class should contain the following two methods:

1. **public** List<Teacher> findSubjectTeacher(List<Teacher> allTeachers, String subjectName) {}

This method input a List of Teacher type as an argument and a String type value and returns a list of Teacher type. This method will find all such teachers in the allTeachers list that have the subject subjectName. Ideally, there should be more than a few teachers for the same subject.

2. **public** Set<Teacher> getBusyTeachers(List<Teacher> allTeachers) {}

This method inputs a List of Teacher type as an argument and returns a Set of Teacher type. A busy teacher is a teacher that has at least 3 or more subjects to teach. If the list contains duplicate elements, make sure that the duplicate items are not returned from the method. Increase the salary of such teachers by 5%

Create a separate execution java class called Main.java and call both the methods given above!

Task 6

Create a java class file called SmartPhone.java. The SmartPhone.java class should contain the following members:

1. a private field to store the IMEI number of the smartphone, as a reference Long type value. **IMMUTABLE**
2. a private field to store the name of the smartphone, as a reference String type value. **IMMUTABLE**
3. a private field to store the price of the smartphone, as a primitive double type value.
4. a private field to store the screen size of the smartphone (in inches), as a primitive byte type value.
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called OnlineShop.java. The class should contain the following two methods:

1. **public** List<SmartPhone> getListOfSmartPhones
(Set<SmartPhone> smartPhones, **byte** requiredScreenSize) {}

This method inputs a Set of SmartPhone type and a required screen size value of byte type. This method will iterate through all the smart phone objects' set and find out the smart phones whose screen size is at least as large as the required screen size value. If you find a phone whose screen size is lesser, decrease its price by 10%

2. **public** List<SmartPhone> findBudgetSmartPhones(Set<SmartPhone> smartPhones) {}

This method inputs a Set of SmartPhone type and returns a List of SmartPhone type. A budget smart phone is a phone whose price is less than or equal to 10,000 INR. If you do not find any such phone in the set of smart phones, make sure that your method does not return null!

Create a separate execution java class called Main.java and call both the methods given above!

Task 7

Create a java class file called Book.java. The Book.java class should contain the following members:

1. a private field to store the ISBN number of the book, as a reference String type value. **IMMUTABLE**
2. a private field to store the name of the book, as a reference String type value. **IMMUTABLE**
3. a private field to store the name of the author of the book, as a reference String type value.
4. a private field to store the price of the book, as a primitive double type value.
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called CollegeLibrary.java. The class should contain the following two methods:

1. **public** List<Book> **getListOfExpensiveBooks**(Set<Book> books) {}

This method input a Set of Book type and returns a List of Book type. An expensive book is a book whose price is at least 5,000 INR or more. If such a book exists, make sure that the returning list should not have duplicate names.

If there are two books by the same author, even with different names, they should be considered as same.

2. **public** Book **findBookDetailsByISBN**(List<Book> books, String ISBN) {}

This method will find a book by the ISBN number and return it. Make sure even if there is not such book, the method should not return null.

Create a separate execution java class called Main.java and call both the methods given above!

Task 8

Create a java class file called Book.java. The Book.java class should contain the following members:

1. a private field to store the ISBN number of the book, as a reference String type value. **IMMUTABLE**
2. a private field to store the name of the book, as a reference String type value. **IMMUTABLE**
3. a private field to store the name of the author of the book, as a reference String type value.
4. a private field to store the subject of the book, as a reference String type value.
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called BooksHelper.java. The class should contain the following two methods:

1. `public List<Book> findAllBooksOnSubject(Set<Book> books, String subject) {}`

This method will find all the books on the specified subject, the returned list should not return null AND it should not have any duplicate values.

2. `public List<Book> sortBooks(Set<Book> books) {}`

This method will sort the set of books provided to this method on the basis of their subjects.

Create a separate execution java class called Main.java and call both the methods given above!

Task 9

Create a java class file called Patient.java. The Patient.java class should contain the following members:

1. a private field to store the patient id, as a primitive short type value.
2. a private field to store the name of the patient, as a reference String type value. **IMMUTABLE**
3. a private field to store the age of the patient, as a primitive byte type value.
4. a private field to store the contact number of the patient, as a primitive long type value. **IMMUTABLE**
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called PatientHelper.java. The class should contain the following two methods:

1. **public** Set<Patient> **getAllOldPatients**(List<Patient> patients) {}

This method will return all the patients whose age is at least 60 or above. If there is no such patient, make sure that the method does not return null. The method should not return duplicate items as well.

2. **public** List<Patient> **sortPatientsAccordingToDecreasingAge**(List<Patient> patients) {}

This method will sort the patients' list according to decreasing age, a patient with a greater age should come first in the list. Use either a lambda expression or a method reference to do this.

Create a separate execution java class called Main.java and call both the methods given above!

Task 10

Create a java class file called Patient.java. The Patient.java class should contain the following members:

1. a private field to store the patient id, as a primitive short type value. **IMMUTABLE**
2. a private field to store the name of the patient, as a reference String type value.
3. a private field to store the result of the patient, as a primitive boolean type value.
4. a private field to store the contact number of the patient, as a primitive long type value. **IMMUTABLE**
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called TestGenerator.java. The class should contain the following two methods:

1. **public** Set<Patient> **getAllPositivePatients**(List<Patient> allPatients) {}

This method will return all the patients whose results have come out as positive. If there are no such patients, make sure that the method does not return null, it should also not return any duplicate values.

2. **public** Patient **findPatient**(List<Patient> patients, **short** patientID) {}

This method will return all the details of a patient by searching for them in the list of patients using their patient ID, if there is no such patient, the method should not return null.

Create a separate execution java class called Main.java and call both the methods given above!

Task 11

Create a java class file called Laptop.java. The Laptop.java class should contain the following members:

1. a private field to store the name of the laptop, as a reference String type value. **IMMUTABLE**
2. a private field to store the ram of the laptop in GBs, as a primitive byte type value.
3. a private field to store the price of the laptop in INR, as a primitive double type value.
4. a private field to indicate the availability of the laptop, as a primitive boolean type value. **IMMUTABLE**
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called LaptopBuyerHelper.java. The class should contain the following two methods:

1. `public List<Laptop> getLaptopList(List<Laptop> laptops, byte ramCapacity) {}`

This method will return all the laptop objects whose ram capacity is at least (or more) than the ram capacity specified in the parameters. A laptop should only be returned from the method if it is available in stock. If no such laptop is available, the method should not return null. The method should also not return duplicates.

2. `public Set<Laptop> findLaptop(List<Laptop> laptops, byte ramCapacity, double price) {}`

This method will return all the laptop objects whose price is at most (or lower) than the price specified in the parameters, if showing a laptop with a different price, make sure the ram should be the same. If no such laptop is available, the method should not return null. The method should also not return duplicates.

Create a separate execution java class called Main.java and call both the methods given above!

Task 12

Create a java class file called Laptop.java. The Laptop.java class should contain the following members:

1. a private field to store the name of the laptop, as a reference String type value. **IMMUTABLE**
2. a private field to store the graphic cards' capacity in GBs, as a primitive byte type value.
3. a private field to store the ram capacity in GBs, as a primitive byte type value.
4. a private field to store the price of the laptop in INR, as a primitive double type value.
5. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called LaptopComparison.java. The class should contain the following two methods:

1. **public** Set<Laptop> getGamingLaptops
 (List<Laptop> laptops, **byte** ramCapacity, **byte**
 graphicsCardCapacity) {}

This method will return all the laptops which have a ram capacity of at least (or more) as specified in the argument, AND a graphics card capacity (or more) as specified in the argument. The returning set from the method should be sorted in a decreasing order of their price, if price is same use RAM, if RAM is same, use graphics.

2. **public** Set<Laptop> sortGamingLaptops(Set<Laptop> laptops) {}

This method will sort the gaming laptops for the getGamingLaptops() method. The set of the laptops should be sorted in a decreasing order of their price.

Create a separate execution java class called Main.java and call both the methods given above!

Task 13

Create a java class file called Question.java. The Question.java class should contain the following members:

1. a private field to store the question ID, as a primitive int type value.
2. a private field to store the text of the question, as a reference String type value.
3. a private field to store the list of options (a, b, c or d), List<String> (List of String) for options.
4. a private field to store the correct options, as a primitive char type value.
5. a private field to store the marks of the question, as a primitive byte type value.
6. the necessary getter/setter methods for the private fields, at least two constructors for this class, the equals(), toString() and the hashCode() methods.

Create a java class file called TestGenerator.java. The class should contain the following two methods:

1. `public HashSet<Question> fetchQuestionList(List<Question> allQuestions, byte marks) {}`

This method will iterate through the allQuestions list and find all the questions that have the same marks as the marks argument, and return a HashSet of the questions, the marks can not be greater than 10, the returning hash set cannot be null, and there should be not duplicate values in the hash set.

2. `public HashSet<Question> createTest(List<Question> allQuestions) {}`

This method will fetch 10 random questions from the list of all questions, but make sure that, the returning set of questions cannot be null, cannot have duplicate elements, and the total sum of the marks of all the questions cannot be greater than 50.

Create a separate execution java class called Main.java and call both the methods given above!