**Java Activity4**

**Collection**

**Question 1.1**

Display unique **5 random numbers. The random number must be between 1 and 15**

**Question** **1.2**

Create the class User

|  |
| --- |
| User |
| userId  pwd  mobile  address |
| //setter &getter |

|  |
| --- |
| <<IUserService>> |
| User doLogin(id,pwd) |

|  |
| --- |
| <<IUserDao>> |
| getUsers() |
|  |

* Maintain the user objects in the map fetched from file or db
* Display the user mobile no and address, if userid and pwd is validated
* Maintain service and dao layer
* Dao implementation can be change from file or db
* Use OO concept and best practices
* If dao implementation is file, the file contains comma separated format

|  |
| --- |
| 1001,ram,9999999999,delhi  1002,tom,8888888888,noida  1003,sam,9888877777,gurgaon |

**Question 1.3**

Create the class Product

|  |
| --- |
| Product |
| prodId  prodName  price  brand  catId |
| //setter & getter |

Write the crud application

* To add a new product
* To remove the product by prodid
* To view all the products in the order of prodId
* To view the product by prodid
* To view the product by catId in the order of price

Write a menu driven code that interacts with dao layer and the code is loosely coupled

Don’t sort at SQl level, instead use comparable and comparator

**Question 1.4**

Find out duplicates from the list

**Collection using Stream**

**Question 2.1**

* Display the numbers in the list that is divisible by 5
* Display each numbers in the list as n\*n
* Sort the elements in the stream
* Combine two different list in to single list using stream
* Find max, min, sum and average value from the stream
* Find the product of all numbers in the list
* Count the numbers in the list that is divisible by 5

**Question 2.2**

Create the class Product 🡪 pid, pname,price,category

Maintain the products in the list

Using streams

* Display the products from the stream in the order of pid
* Display the products whose category is ‘computer’ in the order of price
* Display the products whose price is greater than or equal to 5000
* Count no of products whose category is ‘computer’
* Find the sum of all product price using sum()
* Find the sum of all product price using reduce()
* extracted first 3 characters of product name and collect to a new list