

Generics and Collections
Subject: CSW2(CSE3141)
Session: Feb 2024 to April 2024
Branch: CSE&CSIT
Section : All

1. Create a generic class **Pair<K,V>** with private variables **key** and **value**. The class **Pair** should define a parameterised constructor and getter and setter methods for these attributes. After addition of objects, the **main** class should retrieve and print the pair of **key** and **value**.
2. Write a Java code snippet that comprises of a **User** class and an **ArrayListUser** class. The **User** class should define private fields for **name** and **age**, along with a parameterized constructor and getter/setter methods for these attributes. Create an **ArrayListUser** class of **User** objects. After addition of objects, the **ArrayListUser** class should retrieve and print the name and age of users. Then, it should sort the user according to age using getter methods and print the updated array list of users.
3. Write a Java code snippet that comprises of a **Car** class and a **CarApp** class. The **Car** class should define private fields for **ModalNo(int)**, **name(string)** and **stock(int)**. Define a parameterised constructor and override the **compareTo** method as **public int compareTo(Car car)** to sort the car on basis of the total number of stock. Create an **ArrayList** of **Car** objects and print the updated the sorted list.
For example the list of sorted Car Objects
2013 creta 10
2020 MG 13
2018 Kia 20
2017 Audi 45
2015 BMW 55
4. Create a class **Student** having member variable **name**, **age**, and **mark** and required get and set methods. Create a **LinkedList** of **Student** type and perform the below operation on it.
(a.) Display the list

- (b.) Ask the user to enter a **Student** object and print the existence of the object. Specify if the object is searched according to reference or content.
- (c.) Remove a specified **Student** object.
- (d.) Count the number of objects present in the list.
- (e.) Override **equals** method checking if the two Student objects share all the same values.

5. Create a Class **Book** that has id, name, author and quantity for each book issued. The **Book** class should define a parameterised constructor. Design a class **Library** that creates a HashMap of books which contains an entry of key as **Integer** and value as **Book** object. Instantiate at least two **Book** objects and display the collection of books in the HashMap. Use proper methods of HashMap class to return the following things

- (a.) Check if a particular book name is present in the map
- (b.) Remove the value associated with a particular key value which will remove the book entry.

6. Write a program to create a TreeSet of Integer type and perform the below operation on it.

- (a.) Display the TreeSet
- (b.) Ask the user to enter a number and search that number is it present in

the

list or not.

- (c.) Remove an element from tree.

7. Write a Java code that comprises of a class **Address**, having member variables **plot no**, **at**, **post** and required parameterised constructor. Create a **Tree map** having **key** as **name of a person** and **value** as **address**. Insert required key and value in the created tree map and use an iterator to display it.

8. Find if two strings are anagrams, an anagram is a word or phrase formed by reordering the letters of another word or phrase. Declare two strings **str1** and **str2** and initialize. Create a **HashMap<Character, Integer>** and use methods **containsKey()**, **put()**, **get()** to check the strings.

9. Given an array of integers, print the repeating integers in the array with the help of a HashSet.
10. In given large string, find the most occurring words in the string.
Hint:-
- Create a **Hashtable** which will keep track of <word, frequency>
 - Iterate through the string and keep track of word frequency by inserting into Hash-Table.
 - When we have a new word, we will insert it into the **Hashtable** with frequency 1. For all repetition of the word, we will increase the frequency.
 - We can keep track of the most occurring words whenever we are increasing the frequency we can see if this is the most occurring word or not.
11. Given an unsorted array of integers from 1 to 10, find smallest positive number missing in the array. Use a hash map HashMap<Integer, Integer> to keep track of elements.
12. Declare an array of integers. `int[] arr = {1,2,10,8,7,3,4,6,5,9};`. Then create a min heap of elements from the array using Priority Queue class. Again Dequeue elements of Priority Queue using appropriate methods.