

**Java Fourth and final Assignment**

**Note:-**

* **In this assignment You must have to create method. and a separate class for each question first and then call it in main.**
* **Program programming must take input from user.**
* **Make as generic as possible.**
* **For projects make PUML Sequence diagrams which show relation b/w class and methods.**

**General question (want to see your imagination)**

**And Write Theory also as comment .**

Java Program to Create Abstract Class

Java Program to Create Singleton Class

Java Program to Create an Interface

Java Program to Show Encapsulation in Class

Java Program to Show Inheritance in Class

Java Program to Show Abstraction in Class

Java Program to Show Data Hiding in Class

Java Program to Show Polymorphism in Class

Java Program to Show Overloading of Methods in Class

Java Program to Show Overriding of Methods in Classes

Java Program to Show Use of Super Keyword in Class

Java Program to Show Use of This Keyword in Class

Java Program to Show Usage of Static keyword in Class

Java Program to Show Usage of Access Modifier

**Java collection assignments.**

**Java Collection: Array List**

1. Write a Java program to create an array list, add some colors (strings), and print out the collection.

2. Write a Java program to iterate through all elements in an array list.

3. Write a Java program to insert an element into the array list at the first position.

4. Write a Java program to retrieve an element (at a specified index) from a given array list.

5. Write a Java program to update an array element by the given element.

6. Write a Java program to remove the third element from an array list.

7. Write a Java program to search for an element in an array list.

8. Write a Java program to sort a given array list.

9. Write a Java program to copy one array list into another.

10. Write a Java program to shuffle elements in an array list.

**Java Collection: LinkedList**

1. Write a Java program to append the specified element to the end of a linked list.

2. Write a Java program to iterate through all elements in a linked list.

3. Write a Java program to iterate through all elements in a linked list starting at the specified position.

4. Write a Java program to iterate a linked list in reverse order.

5. Write a Java program to insert the specified element at the specified position in the linked list.

6. Write a Java program to insert elements into the linked list at the first and last positions.

7. Write a Java program to insert the specified element at the front of a linked list.

8. Write a Java program to insert the specified element at the end of a linked list.

9. Write a Java program to insert some elements at the specified position into a linked list.

10. Write a Java program to get the first and last occurrence of the specified elements in a linked list.

**Java Collection: HashSet Exercises**

1. Write a Java program to append the specified element to the end of a hash set.

2. Write a Java program to iterate through all elements in a hash list.

3. Write a Java program to get the number of elements in a hash set.

4. Write a Java program to empty an hash set.

5. Write a Java program to test if a hash set is empty or not.

**Java Collection: TreeSet Exercises**

1. Write a Java program to create a tree set, add some colors (strings) and print out the tree set.

2. Write a Java program to iterate through all elements in a tree set.

3. Write a Java program to add all the elements of a specified tree set to another tree set.

4. Write a Java program to create a reverse order view of the elements contained in each tree set.

5. Write a Java program to get the first and last elements in a tree set.

**Java Collection: Priority Queue**

1. Write a Java program to create a priority queue, add some colors (strings) and print out the elements of the priority queue.

2. Write a Java program to iterate through all elements in the priority queue.

3. Write a Java program to add all the elements of a priority queue to another priority queue.

4. Write a Java program to insert a given element into a priority queue.

5. Write a Java program to remove all elements from a priority queue.

**Java Collection: HashMap Exercises**

1. Write a Java program to associate the specified value with the specified key in a HashMap.

2. Write a Java program to count the number of key-value (size) mappings in a map.

3. Write a Java program to copy all mappings from the specified map to another map.

4. Write a Java program to remove all mappings from a map.

5. Write a Java program to check whether a map contains key-value mappings (empty) or not.

**Java Collection: Tree Map Exercises**

1. Write a Java program to associate the specified value with the specified key in a Tree Map.

2. Write a Java program to copy Tree Map's content to another Tree Map.

3. Write a Java program to search for a key in a Tree Map.

4. Write a Java program to search for a value in a Tree Map.

5. Write a Java program to get all keys from a Tree Map.

**Java Streams**

1. Write a Java program to calculate the average of a list of integers using streams.

2. Write a Java program to convert a list of strings to uppercase or lowercase using streams.

3. Write a Java program to calculate the sum of all even, odd numbers in a list using streams.

4. Write a Java program to remove all duplicate elements from a list using streams.

5. Write a Java program to count the number of strings in a list that start with a specific letter using streams.

**Java thread**

1. Write a Java program to create a basic Java thread that prints "Hello, World!" when executed.

2. Write a Java program that creates two threads to find and print even and odd numbers from 1 to 20.

3. Write a Java program that sorts an array of integers using multiple threads.

4. Write a Java program that performs matrix multiplication using multiple threads.

5. Write a Java program that calculates the sum of all prime numbers up to a given limit using multiple threads.

6. Write a Java program to implement a concurrent web crawler that crawls multiple websites simultaneously using threads.

7. Write a Java program that creates a bank account with concurrent deposits and withdrawals using threads.

**Java Multithreading**

**1.** Write a Java program to create and start multiple threads that increment a shared counter variable concurrently.

**2.** Write a Java program to create a producer-consumer scenario using the wait() and notify() methods for thread synchronization.

**3.** Write a Java program that uses the ReentrantLock class to synchronize access to a shared resource among multiple threads.

**4.** Write a Java program to demonstrate Semaphore usage for thread

**5.** Write a Java program to showcase the usage of the CyclicBarrier class for thread synchronization.

**6.** Write a Java program that uses the CountDownLatch class to synchronize the start and finish of multiple threads.

**7.** Write a Java program to illustrate the usage of the ReadWriteLock interface for concurrent read-write access to a shared resource.

**8.** Write a Java program demonstrating how to access a map concurrently using the ConcurrentHashMap class.

**9.** Write a Java program that utilizes the ConcurrentLinkedQueue class to implement a thread-safe queue.

**10.** Write a Java program to showcase the usage of the Phaser class for coordination and synchronization of multiple threads.

**11.** Write a Java program that utilizes the Exchanger class for exchanging data between two threads.

**12.** Write a Java program to demonstrate the usage of the Callable and Future interfaces for executing tasks asynchronously and obtaining their results.

**13.** Write a Java program that uses the ScheduledExecutorService interface to schedule tasks for execution at a specified time or with a fixed delay.

**14.** Write a Java program to demonstrate the usage of the ForkJoinPool class for parallel execution of recursive tasks.

**15.** Write a Java program that utilizes the StampedLock class for optimizing concurrent read-write access to a shared resource.

**Java input-output**

Write a Java program to get a list of all file/directory names in the given directory.

Write a Java program to get specific files with extensions from a specified folder.

Write a Java program to check if a file or directory specified by pathname exists or not.

Write a Java program to check if a file or directory has read and write permissions.

Write a Java program to check if the given pathname is a directory or a file

**Java exercises on exception handling**

* + Write a Java program to create a method that takes an integer as a parameter and throws an exception if the number is odd.
  + Write a Java program to create a method that reads a file and throws an exception if the file is not found.
  + Write a Java program that reads a list of numbers from a file and throws an exception if any of the numbers are positive.
  + Write a Java program that reads a file and throws an exception if the file is empty.
  + Write a Java program that reads a list of integers from the user and throws an exception if any numbers are duplicates.
  + Write a Java program to create a method that takes a string as input and throws an exception if the string does not contain vowels.

**Java Lambda**

1. Write a Java program to implement a lambda expression to find the sum of two integers.

2. Write a Java program to implement a lambda expression to check if a given string is empty.

3. Write a Java program to implement a lambda expression to convert a list of strings to uppercase and lowercase.

4. Write a Java program to implement a lambda expression to filter out even and odd numbers from a list of integers.

5. Write a Java program to implement a lambda expression to sort a list of strings in alphabetical order.

6. Write a Java program to implement a lambda expression to find the average of a list of doubles.

7. Write a Java program to implement a lambda

expression to remove duplicates from a list of integers.

8. Write a lambda expression to implement a lambda expression to calculate the factorial of a given number.

9. Write a Java program to implement a lambda expression to create a lambda expression to

10. Write a Java program to implement a lambda expression to concatenate two strings.

**Java Regular**

Write a Java program that takes a number and sets a thousand separators for that number.

Write a Java program to remove all non-alphanumeric characters from a given string.

Write a Java program to validate a phone number.

Write a Java program to move all lower-case letters to the front of a given word. This will keep the relative position of all the letters (both upper and lower case) same.

Write a Java program to separate consonants and vowels from a given string.

Write a Java program to get the last n vowels of a given string.

**Oops and collection concept-based minor project**

**Note: -**

**In each project used all oops concept. (programmer can decide how they can show)**

**In project use the interface and abstract class to call the business logic class and then call it in main class.**

**Make sequence diagram of each.**

**Online Shopping System:**

Develop an online shopping platform with classes like Product, Customer, and Shopping Cart.

Implement functionalities for adding/removing items, processing orders, and managing customer accounts.

Utilize inheritance for different product categories and encapsulation for secure customer data.

**School Management System:**

Create a system to manage a school's data with classes like Student, Teacher, and Course.

Implement features for enrolling students in courses, managing teacher schedules, and recording grades.

Create a StudentList class that manages a list of students. Include methods to add, remove, and display student details.

Use inheritance for different types of courses or departments and encapsulation for data privacy.

**Hotel Management System:**

Develop a system to manage hotel bookings and services with classes like Room, Guest, and Reservation.

Implement functionalities for room booking, checking availability, billing, and guest management.

Use inheritance for different room types or amenities and encapsulation for secure guest information.

**Inventory Management System:**

Create a system to handle inventory with classes like Product, Warehouse, and Inventory Manager.

Implement features for adding/removing items, tracking stock levels, and managing orders.

Utilize inheritance for different product categories and encapsulation for secure inventory data.