Task 1:

Task 1 : Write a Java program to copy the alternative elements in the array.

Task 2 : Write a java program to accept the students name and marks in subjects and find the total and average of the marks and sort the list accordingly.

Task 3 : Demonstrate with an example with the difference between equals() vs deepEquals() method?

Task 4 : Use ArrayCopyRange() function to demonstrate and example such that you copy the range of values in an array.

Task 5 : You are required to create a string and captialize every alternative character. for
eg : if the input is anurag then output should be AnUrAg

Task 2:

Create a class named Employee with the following details:

* empId :: int
* empName :: String
* sal :: double

Methods

getEmployeeDetails()
setEmployeeDetails()
getLoanEligibility()

check if the employee is eligible to get loan. The conditions arE:

* an employee should have worked for greater than 5 years.
* If his /her annual salary is 6 lakhs then 2 lakhs of loan is granted.
* If his/her annual salary is >=10 lakhs then 5 lakhs of loan is granted.
* If his/her annual salary is >=15 lakhs then 7 lakhs of loan is granted.

Task 3 - IO Streams:

1. Write a Java program to get a list of all file/directory names in the given directory.
2. Write a Java program to get specific files with extensions from a specified folder.
3. Write a Java program to check if a file or directory specified by pathname exists or not.
4. Write a Java program to check if a file or directory has read and write permissions.
5. Write a Java program to check if the given pathname is a directory or a file.
6. Write a Java program to determine the last modified date of a file.
7. Write a Java program to read input from the Java console.
8. Write a Java program to get the file size in bytes, KB, MB.
9. Write a Java program to read the contents of a file into a byte array.
10. Write a Java program to read file content line by line.
11. Write a Java program to read a plain text file.

Task 4 - Exception Handling:

1. Write a Java program that throws an exception and catch it using a try-catch block.
2. Write a Java program to create a method that takes an integer as a parameter and throws an exception if the number is odd.
3. Write a Java program to create a method that reads a file and throws an exception if the file is not found.
4. Write a Java program that reads a list of numbers from a file and throws an exception if any of the numbers are positive.
5. Write a Java program that reads a file and throws an exception if the file is empty.

Task 5 - Threads:

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

Task 6 - Generics:

Write a Java program to create a generic method that takes two arrays of the same type and checks if they have the same elements in the same order.

1. Write a Java program to create a generic method that takes a list of numbers and returns the sum of all the even and odd numbers.
2. Write a Java program to create a generic method that takes a list of any type and a target element. It returns the index of the first occurrence of the target element in the list. Return -1 if the target element cannot be found.
3. Write a Java program to create a generic method that takes a list of any type and returns it as a new list with the elements in reverse order.
4. Write a Java program to create a generic method that takes two lists of the same type and merges them into a single list. This method alternates the elements of each list.

Task 7 - Collection Api:

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 shufle elements in an array list.
11. Write a Java program to append the specified element to the end of a linked list.
12. Write a Java program to iterate through all elements in a linked list.
13. Write a Java program to iterate through all elements in a linked list starting at the specified position.
14. Write a Java program to iterate a linked list in reverse order.
15. Write a Java program to insert the specified element at the specified position in the linked list.
16. Write a Java program to insert elements into the linked list at the first and last positions.
17. Write a Java program to insert the specified element at the front of a linked list.
18. Write a Java program to insert the specified element at the end of a linked list.
19. Write a Java program to insert some elements at the specified position into a linked list.
20. Write a Java program to get the first and last occurrence of the specified elements in a linked list.

Task 8 - JDBC:

Task 1 : Recursively insert multiple row/s into the table.

Hint :: Use PreparedStatement

Task 2 : Update the row of a table dynamically using Scanner class.

Task 3 : Delete the row from the table by using a delete query using Scanner class. Task 4 : Create a menu driven program where the options would be

1. Insert a new Row
2. Update a Row
3. Delete a Row
4. Select the row
5. Exit

Use PreparedStatement class for all the above tasks.

Task 9 - 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.
6. Write a Java program to sort a list of strings in alphabetical order, ascending and descending using streams.
7. Write a Java program to find the maximum and minimum values in a list of integers using streams.

Task 10:

Given a table with rows and columns. Perform CRUD operatios on it using JDBC API

Create a Java program to represent Interthread communication

Create an object called student in Javascript with 3 properties and 1 method and access them with the object.
Then add a nested object and 2 more functions to it. Access all the properties and functions.
Remove a normal property and a function and then display the details of the object.

Create a simple Java servlet to display a HTML file with its content.