

Note:

All the questions are compulsory.

The duration of the test is 3 hours

Mode of Submission: Github

Link Don't seek help from any person/resource during the test.

Marks Distribution is as follows:

Q1: 2 Marks

Q2: 1.5 Marks

Q3: 2 Marks

Q4: 2 Marks

Q5: 1.5 Marks

Q6: 1 Marks

Q1/-Create a multithreaded Producer-Consumer application, where the Producer Thread will produce a number. And the Consumer thread should print the sum of all the numbers till that produced number starts from 0.

Ex:

Producer Produced : 0

Consumer Calculated Sum is: 0

Producer Produced: 1

Consumer Calculated Sum is: 1

Producer Produced: 2

Consumer Calculated Sum is: 3

Producer Produced : 3

Consumer Calculated Sum is: 6

Producer Produced: 4

Consumer Calculated Sum is: 10

Producer Produced: 5

Consumer Calculated Sum is: 15

Q2/- Create a Thread Pool of 3 Threads, and submit the 6 tasks using the Callable interface. Each task should accept a Student object (roll, name, address, marks)through the constructor and return the result of that student with his roll number.

Ex:

If supplied Student object is : **new Student(10, "Varun", "pune", 450);**
Then the thread should return: **"Roll 10 student is Fail"**

Q3/- Write a multithreaded application that should perform the following tasks:

- Start the 3 threads.
- Give the name of those threads as Virat, Dhoni, and Rohit.
- Set the max priority to the Dhoni thread.
- Virat Thread should print the Addition of the first 20 numbers.
- Dhoni thread should print from 1 to 10.
- Rohit thread should print the product of 1st 10 number

Note: Make sure that first, the Dhoni thread should print then Rohit Thread, and then Virat Thread.

Q4/- Consider the following Bean classes:

```
class Address{
    state: String
    city: String
    pincode: String
}
```

```
class Employee{

    empId: int
    empName: String
    salary: int
    email: String
    password: String
    address: Address //has-A relationship

}
```

Inside the Demo1 class Serialize the List of 5 Employee objects with all details inside a file called “employeeedata.txt”

- **Make sure that the employee’s password should not be serialized.**
- **Inside the Demo2 class Deserialize the List of Employees from the “employeeedata.txt”**
- **After Deserialization, add one more Employee object inside that List and Serialize it back to the “employeeedata.txt” file.**

Q5: Write a Java program to read a text file using BufferedReader class and write those text to another file using PrintWriter class.

Q6: What is the difference between Runnable and Callable interface?