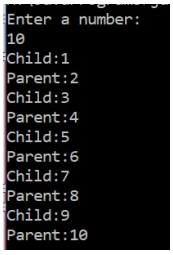
# Assignment 8

**Topic:** Multi-Threading

| Sl. No. | Question |
|---|---|
| 1. | Write a java program that will create one child thread. The child thread has to display all odd numbers between m and n, and the main thread will display all the even numbers between m and n. The sample output is shown below:<br>  |
| 2. | Write a java program to create two threads. First thread should find the square of the number, second thread should find the sum of the digits of the squared number. |
| 3. | Write a java program that will compute product of two vector (1D array) using multithreading. The program should read two vectors (of same size) from the user. First thread should multiply the corresponding elements present in the odd index position and second thread should multiply the corresponding elements present in the even index position. Main thread should display the result. |
| 4. | Write a simple Java thread program to compute the sum of *n* natural numbers. The program should read the number of threads m and value of n from the user. Each of the threads should add its share of assigned number to a global variable. When all the threads are done, the global variable should contain the result. The programshould use a Synchronized block to make sure that only one thread is updating the global variable at a given time. |
| 5. | Write a Java thread program to search the minimum number in a given array. The program should read the number of elements in the array, number of threads to be created and the array elements from the user. Each thread should find minimum element in an assigned block of elements and compare to global minimum element. When all the threads are done, the global variable should contain the minimum element. It should use a Synchronized block to make sure that only one thread is updating the global minimum variable at any given time |
| 6. | Write a java program in which main thread should create two child threads (Producer and Consumer). First child thread (Producer) should produce ten random integers between 1 to 100 and the second child thread (Consumer) should check whether the generated number is even or odd. At the end the second child thread (Consumer) should print total number of even numbers received. Both the threads should wait and notify each other wherever necessary. The sample output is shown below: |