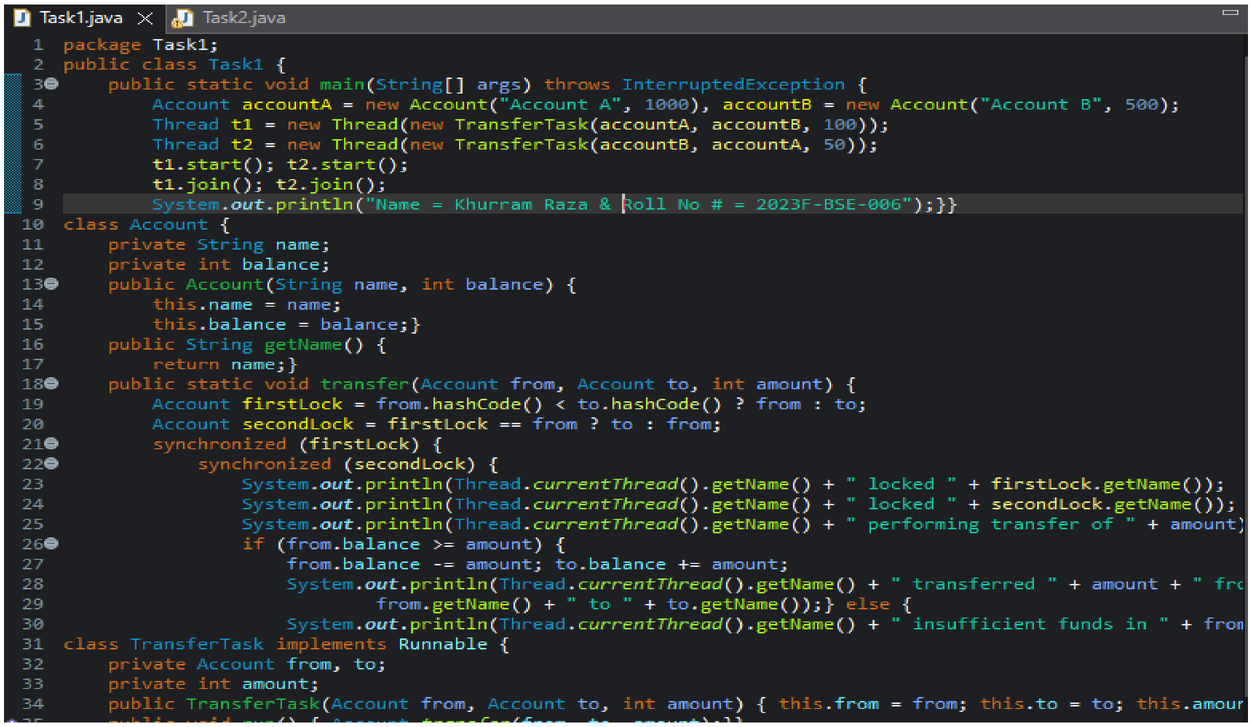
LAB # 7 (OPEN-ENDED LAB)

Lab Task:

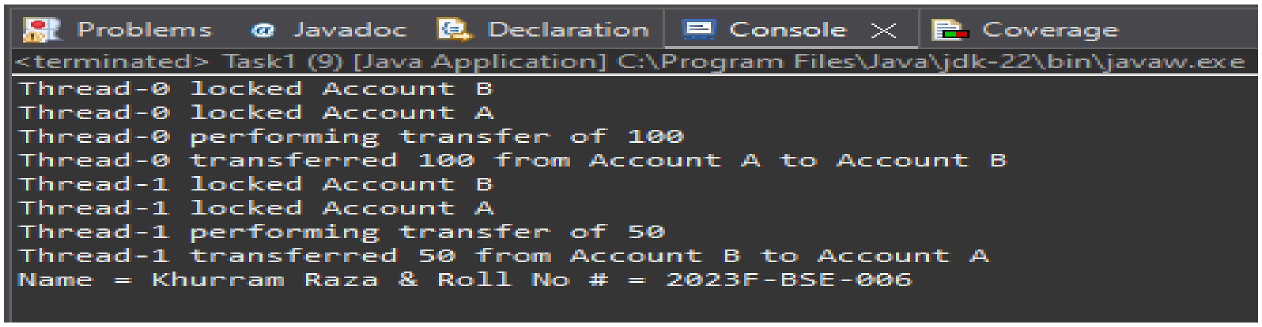
Q.1E: -Create a Java program with two threads performing money transfers between two shared accounts. Use synchronized blocks to acquire locks in the same order to avoid deadlock. Print messages when threads try to lock, acquire locks, perform transfer, and release locks.Simulate Thread 1 transferring $100 from A to B and Thread 2 transferring $50 from B to A.

Code:



```java
package Task1;
public class Task1 {
    public static void main(String[] args) throws InterruptedException {
        Account accountA = new Account("Account A", 1000), accountB = new Account("Account B", 500);
        Thread t1 = new Thread(new TransferTask(accountA, accountB, 100));
        Thread t2 = new Thread(new TransferTask(accountB, accountA, 50));
        t1.start(); t2.start();
        t1.join(); t2.join();
        System.out.println("Name = Khurram Raza & Roll No # = 2023F-BSE-006");}}
class Account {
    private String name;
    private int balance;
    public Account(String name, int balance) {
        this.name = name;
        this.balance = balance;}
    public String getName() {
        return name;}
    public static void transfer(Account from, Account to, int amount) {
        Account firstLock = from.hashCode() < to.hashCode() ? from : to;
        Account secondLock = firstLock == from ? to : from;
        synchronized (firstLock) {
            synchronized (secondLock) {
                System.out.println(Thread.currentThread().getName() + " locked " + firstLock.getName());
                System.out.println(Thread.currentThread().getName() + " locked " + secondLock.getName());
                System.out.println(Thread.currentThread().getName() + " performing transfer of " + amount)
                if (from.balance >= amount) {
                    from.balance -= amount; to.balance += amount;
                    System.out.println(Thread.currentThread().getName() + " transferred " + amount + " fr
                            from.getName() + " to " + to.getName());} else {
                    System.out.println(Thread.currentThread().getName() + " insufficient funds in " + fro
class TransferTask implements Runnable {
    private Account from, to;
    private int amount;
    public TransferTask(Account from, Account to, int amount) { this.from = from; this.to = to; this.amou
```

Output:



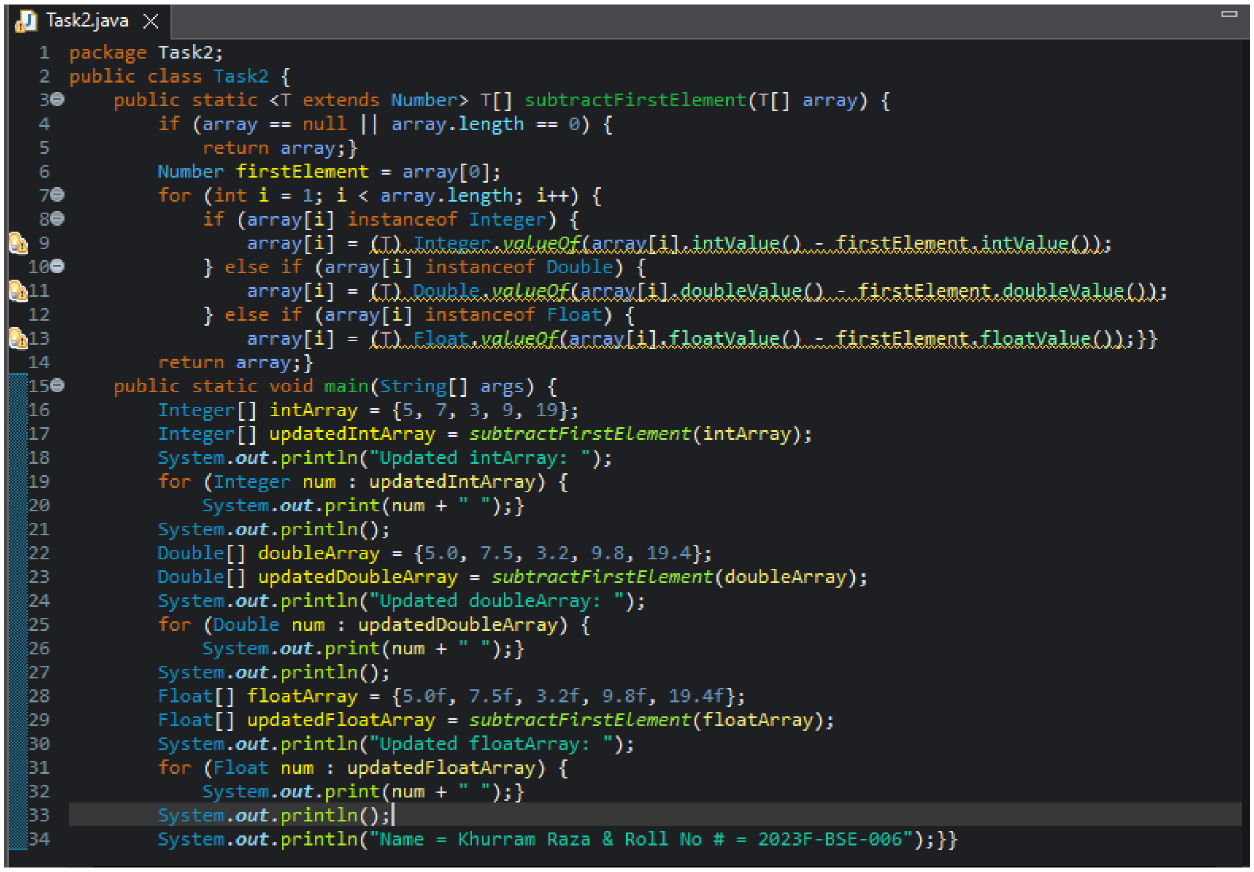
```
<terminated> Task1 (9) [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe
Thread-0 locked Account B
Thread-0 locked Account A
Thread-0 performing transfer of 100
Thread-0 transferred 100 from Account A to Account B
Thread-1 locked Account B
Thread-1 locked Account A
Thread-1 performing transfer of 50
Thread-1 transferred 50 from Account B to Account A
Name = Khurram Raza & Roll No # = 2023F-BSE-006
```

Q.2F: -Write a Java program that takes three arrays: integer array, double array, and float array. Create a generic function that subtracts the first element from all other elements of the array and returns the resulting array:

Input: intArray = [5, 7, 3, 9, 19]

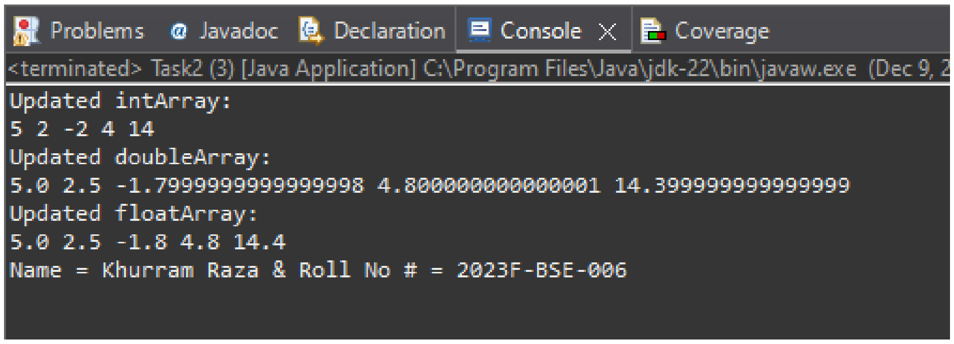
Output: intArray = [0, 2, -2, 4, 14] // 5-5=0, 7-5=2 .

Code:



```java
package Task2;
public class Task2 {
    public static <T extends Number> T[] subtractFirstElement(T[] array) {
        if (array == null || array.length == 0) {
            return array;}
        Number firstElement = array[0];
        for (int i = 1; i < array.length; i++) {
            if (array[i] instanceof Integer) {
                array[i] = (T) Integer.valueOf(array[i].intValue() - firstElement.intValue());
            } else if (array[i] instanceof Double) {
                array[i] = (T) Double.valueOf(array[i].doubleValue() - firstElement.doubleValue());
            } else if (array[i] instanceof Float) {
                array[i] = (T) Float.valueOf(array[i].floatValue() - firstElement.floatValue());}}
        return array;}
    public static void main(String[] args) {
        Integer[] intArray = {5, 7, 3, 9, 19};
        Integer[] updatedIntArray = subtractFirstElement(intArray);
        System.out.println("Updated intArray: ");
        for (Integer num : updatedIntArray) {
            System.out.print(num + " ");}
        System.out.println();
        Double[] doubleArray = {5.0, 7.5, 3.2, 9.8, 19.4};
        Double[] updatedDoubleArray = subtractFirstElement(doubleArray);
        System.out.println("Updated doubleArray: ");
        for (Double num : updatedDoubleArray) {
            System.out.print(num + " ");}
        System.out.println();
        Float[] floatArray = {5.0f, 7.5f, 3.2f, 9.8f, 19.4f};
        Float[] updatedFloatArray = subtractFirstElement(floatArray);
        System.out.println("Updated floatArray: ");
        for (Float num : updatedFloatArray) {
            System.out.print(num + " ");}
        System.out.println();
        System.out.println("Name = Khurram Raza & Roll No # = 2023F-BSE-006");}}
```

Output:



```
<terminated> Task2 (3) [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (Dec 9, 2
Updated intArray:
5 2 -2 4 14
Updated doubleArray:
5.0 2.5 -1.7999999999999998 4.800000000000001 14.399999999999999
Updated floatArray:
5.0 2.5 -1.8 4.8 14.4
Name = Khurram Raza & Roll No # = 2023F-BSE-006
```

GitHub Screenshot: -