

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:

```
Task1.java × Task2.java
1 package Task1;
2 public class Task1 {
3     public static void main(String[] args) throws InterruptedException {
4         Account accountA = new Account("Account A", 1000), accountB = new Account("Account B", 500);
5         Thread t1 = new Thread(new TransferTask(accountA, accountB, 100));
6         Thread t2 = new Thread(new TransferTask(accountB, accountA, 50));
7         t1.start(); t2.start();
8         t1.join(); t2.join();
9         System.out.println("Name = Khurram Raza & Roll No # = 2023F-BSE-006");
10    class Account {
11        private String name;
12        private int balance;
13        public Account(String name, int balance) {
14            this.name = name;
15            this.balance = balance;
16        public String getName() {
17            return name;
18        public static void transfer(Account from, Account to, int amount) {
19            Account firstLock = from.hashCode() < to.hashCode() ? from : to;
20            Account secondLock = firstLock == from ? to : from;
21            synchronized (firstLock) {
22                synchronized (secondLock) {
23                    System.out.println(Thread.currentThread().getName() + " locked " + firstLock.getName());
24                    System.out.println(Thread.currentThread().getName() + " locked " + secondLock.getName());
25                    System.out.println(Thread.currentThread().getName() + " performing transfer of " + amount);
26                    if (from.balance >= amount) {
27                        from.balance -= amount; to.balance += amount;
28                        System.out.println(Thread.currentThread().getName() + " transferred " + amount + " fro
29                            from.getName() + " to " + to.getName());} else {
30                        System.out.println(Thread.currentThread().getName() + " insufficient funds in " + from
31    class TransferTask implements Runnable {
32        private Account from, to;
33        private int amount;
34        public TransferTask(Account from, Account to, int amount) { this.from = from; this.to = to; this.amou
35        public void run() { transfer(from, to, amount);}}
```

Output:

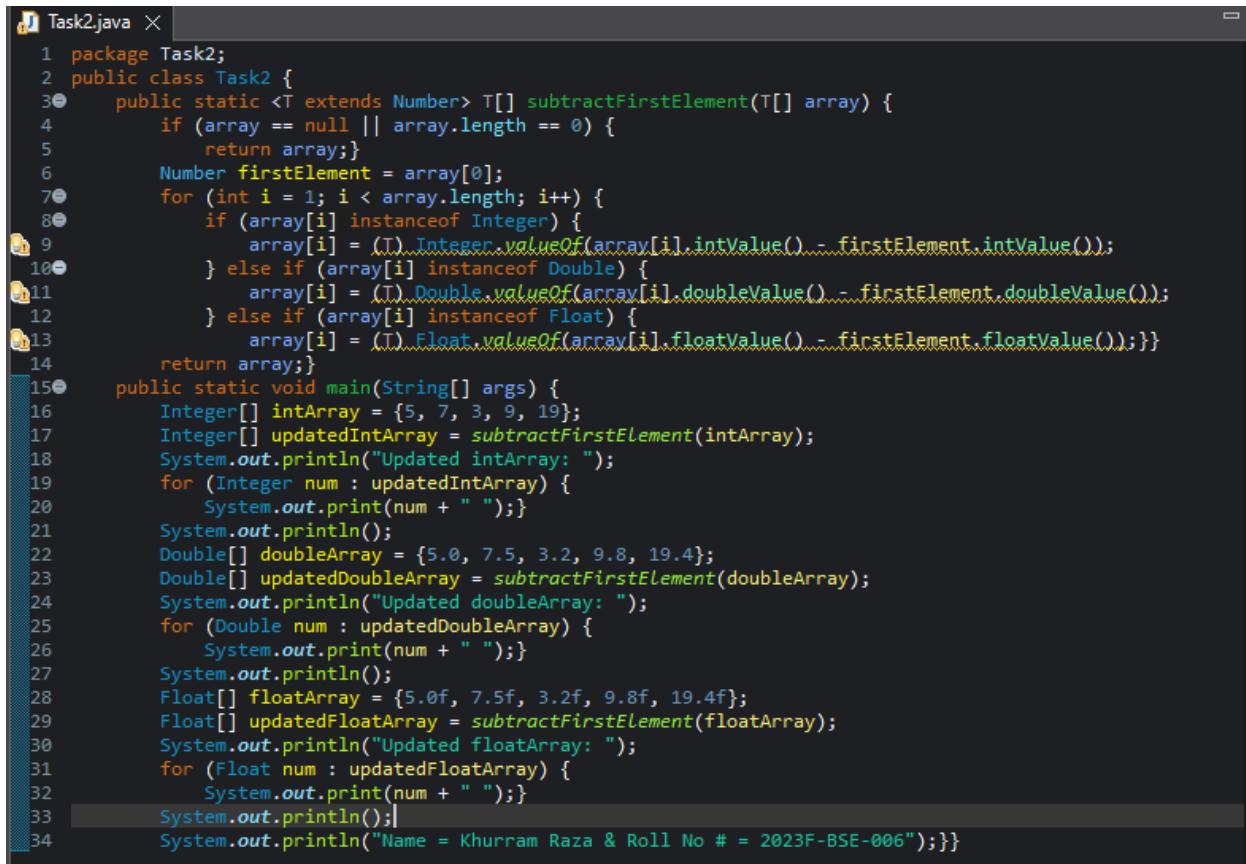
```
Problems @ Javadoc Declaration Console X Coverage
<terminated> Task1 (9) [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe
Thread-0 locked Account B
Thread-0 locked Account A
Thread-0 performing transfer of 100
Thread-0 transferred 100 from Account A to Account B
Thread-1 locked Account 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:

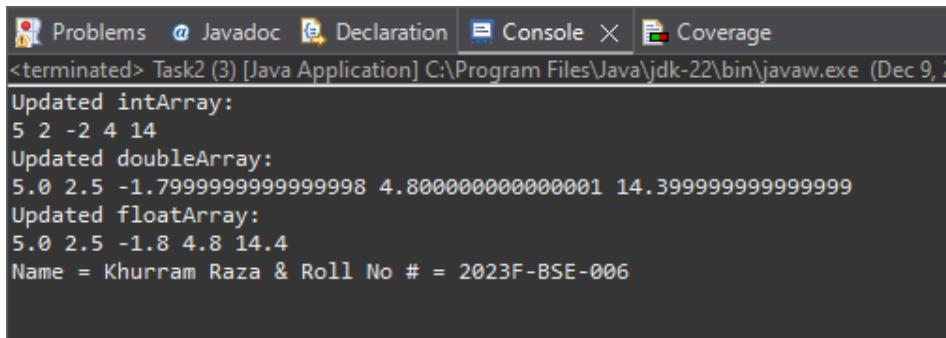


```

1 package Task2;
2 public class Task2 {
3     public static <T extends Number> T[] subtractFirstElement(T[] array) {
4         if (array == null || array.length == 0) {
5             return array;
6         }
7         Number firstElement = array[0];
8         for (int i = 1; i < array.length; i++) {
9             if (array[i] instanceof Integer) {
10                 array[i] = ((Integer) array[i]).intValue() - firstElement.intValue();
11             } else if (array[i] instanceof Double) {
12                 array[i] = ((Double) array[i]).doubleValue() - firstElement.doubleValue();
13             } else if (array[i] instanceof Float) {
14                 array[i] = ((Float) array[i]).floatValue() - firstElement.floatValue();
15             }
16         }
17         return array;
18     }
19     public static void main(String[] args) {
20         Integer[] intArray = {5, 7, 3, 9, 19};
21         Integer[] updatedIntArray = subtractFirstElement(intArray);
22         System.out.println("Updated intArray: ");
23         for (Integer num : updatedIntArray) {
24             System.out.print(num + " ");
25         }
26         System.out.println();
27         Double[] doubleArray = {5.0, 7.5, 3.2, 9.8, 19.4};
28         Double[] updatedDoubleArray = subtractFirstElement(doubleArray);
29         System.out.println("Updated doubleArray: ");
30         for (Double num : updatedDoubleArray) {
31             System.out.print(num + " ");
32         }
33         System.out.println();
34         System.out.println("Name = Khurram Raza & Roll No # = 2023F-BSE-006");
35     }
36 }

```

Output:



```

Problems @ Javadoc Declaration Console Coverage
<terminated> Task2 (3) [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (Dec 9, 2023)
Updated intArray:
5 2 -2 4 14
Updated doubleArray:
5.0 2.5 -1.7999999999999998 4.8000000000000001 14.399999999999999
Updated floatArray:
5.0 2.5 -1.8 4.8 14.4
Name = Khurram Raza & Roll No # = 2023F-BSE-006

```

GitHub Screenshot: -

The screenshot shows a GitHub repository page. At the top, there are navigation links: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the header, the repository name 'Software-Construction-Development-Labs' is displayed, along with a 'Public' badge, a 'Pin' button, a 'Watch' button (0), a 'Fork' button (0), and a 'Star' button (0). A search bar and a code editor interface ('Code') are also visible.

The main content area displays a list of commits:

Author	Commit Message	Date	Commits
KhurramRaza13	Add files via upload	ffe75f3 · now	9 Commits
	README.md	Initial commit	last month
	SCD Lab # 1.pdf	Add files via upload	3 weeks ago
	SCD Lab # 2.pdf	Add files via upload	3 weeks ago
	SCD Lab # 3.pdf	Add files via upload	3 weeks ago
	SCD Lab # 4.pdf	Add files via upload	last month
	SCD Lab # 5.pdf	Add files via upload	3 weeks ago
	SCD Lab # 6.pdf	Add files via upload	3 weeks ago
	SCD Lab # 7 & (OEL).pdf	Add files via upload	now
	SCD Lab # 8.pdf	Add files via upload	2 weeks ago

On the right side, there are sidebar sections for 'Good Programming Practice (E.g Start, Sleep and Stop methods of multithreading)', 'Readme', 'Activity', '0 stars', '0 watching', '0 forks', 'Releases' (No releases published, Create a new release), 'Packages' (No packages published), and 'About'.