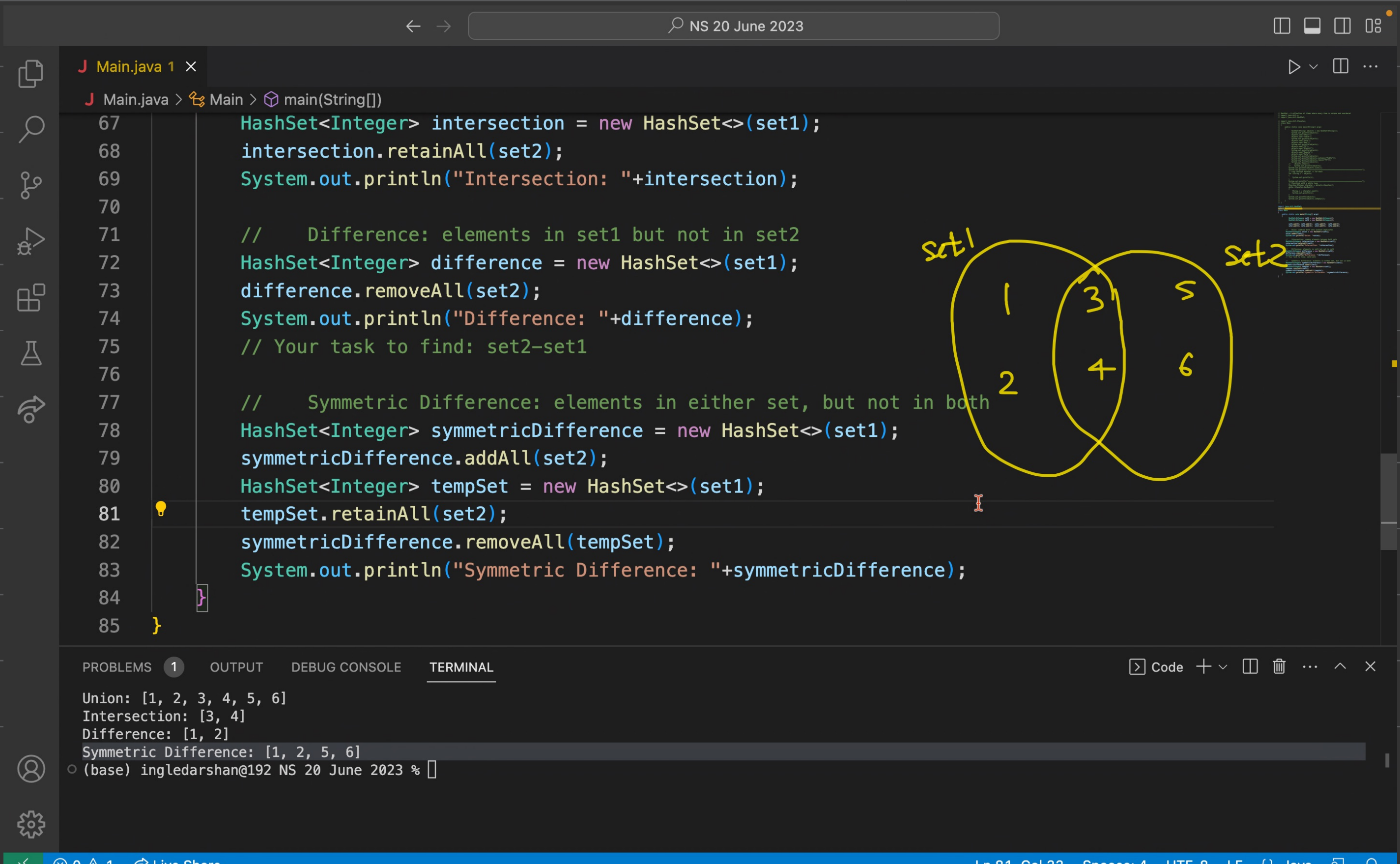
20 June 2023

HashSet: Time Complexity

	Hashset	Array	Sorted Array
Insert / Add :	O (1)	O (1)	O (n)
Search / contains :	O (1)	O (n)	$O(\log_2 n)$
Delete / Remove :	O (1)	O (n)	O (n)

Real time Usage: Java Projects

1. Eliminating duplicates
2. Fast membership checks
3. Set operations: union, intersection and difference - find common elements, unique elements between sets
4. Implementing caches and lookup tables



```java
        HashSet<Integer> intersection = new HashSet<>(set1);
        intersection.retainAll(set2);
        System.out.println("Intersection: "+intersection);

        //    Difference: elements in set1 but not in set2
        HashSet<Integer> difference = new HashSet<>(set1);
        difference.removeAll(set2);
        System.out.println("Difference: "+difference);
        // Your task to find: set2-set1

        //    Symmetric Difference: elements in either set, but not in both
        HashSet<Integer> symmetricDifference = new HashSet<>(set1);
        symmetricDifference.addAll(set2);
        HashSet<Integer> tempSet = new HashSet<>(set1);
        tempSet.retainAll(set2);
        symmetricDifference.removeAll(tempSet);
        System.out.println("Symmetric Difference: "+symmetricDifference);
    }
}
```

```
Union: [1, 2, 3, 4, 5, 6]
Intersection: [3, 4]
Difference: [1, 2]
Symmetric Difference: [1, 2, 5, 6]
(base) ingledarshan@192 NS 20 June 2023 % 
```