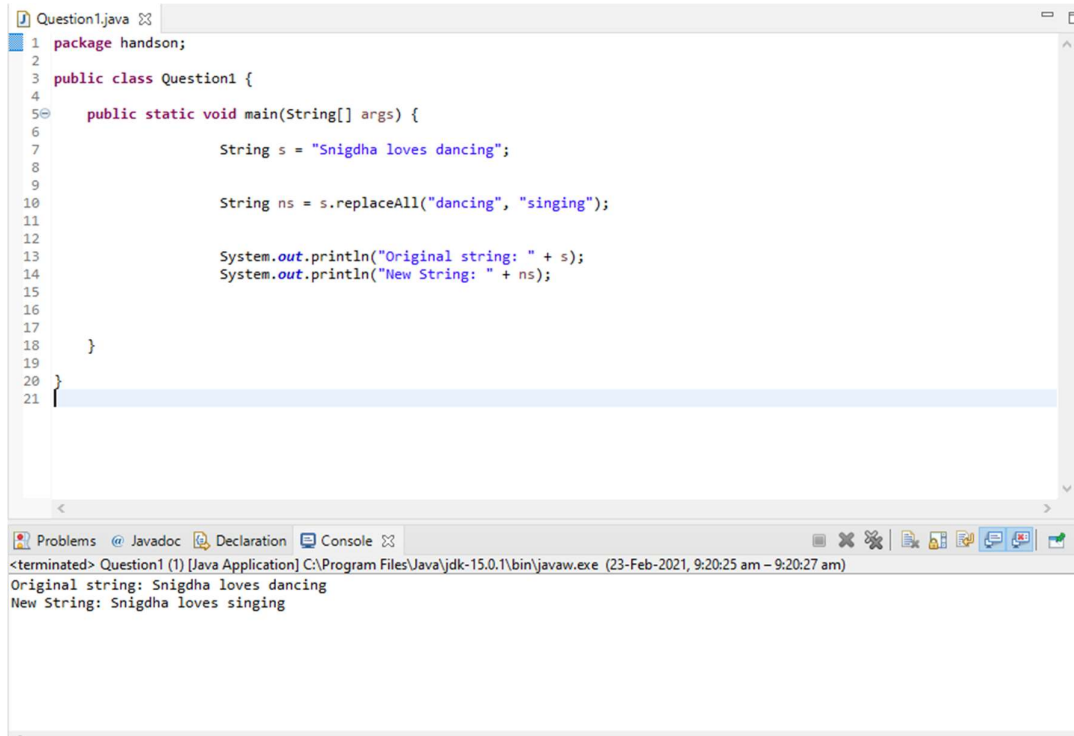


# Java Solutions

## Hands on Section:

1.



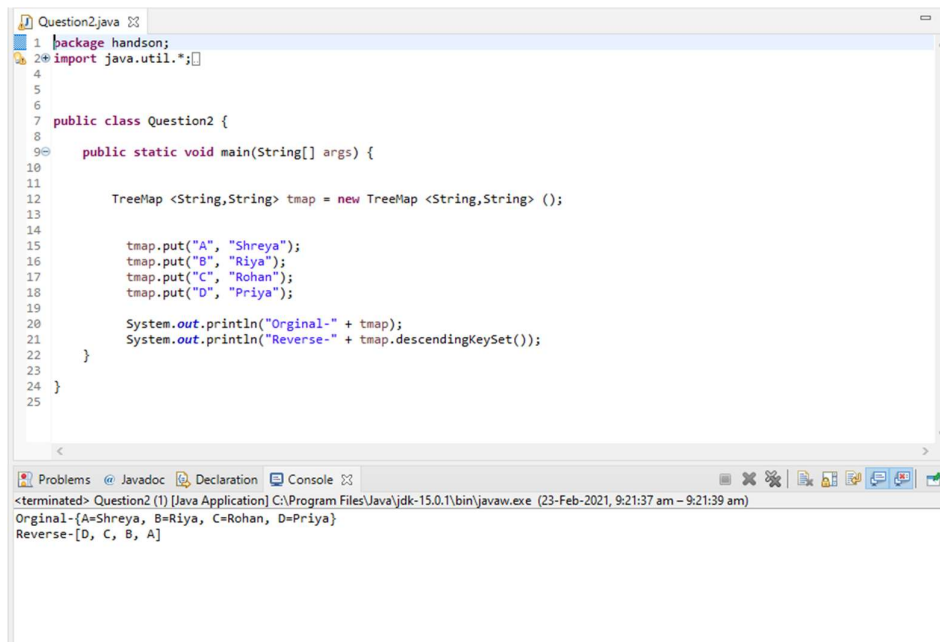
```
1 package handson;
2
3 public class Question1 {
4
5     public static void main(String[] args) {
6
7         String s = "Snigdha loves dancing";
8
9
10        String ns = s.replaceAll("dancing", "singing");
11
12
13        System.out.println("Original string: " + s);
14        System.out.println("New String: " + ns);
15
16
17    }
18 }
19
20
21
```

Problems @ Javadoc Declaration Console

<terminated> Question1 (1) [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (23-Feb-2021, 9:20:25 am – 9:20:27 am)

Original string: Snigdha loves dancing  
New String: Snigdha loves singing

2.



```
1 package handson;
2 import java.util.*;
3
4
5
6
7 public class Question2 {
8
9     public static void main(String[] args) {
10
11
12        TreeMap <String,String> tmap = new TreeMap <String,String> ();
13
14
15        tmap.put("A", "Shreya");
16        tmap.put("B", "Riya");
17        tmap.put("C", "Rohan");
18        tmap.put("D", "Priya");
19
20        System.out.println("Original-" + tmap);
21        System.out.println("Reverse-" + tmap.descendingKeySet());
22    }
23 }
24
25
```

Problems @ Javadoc Declaration Console

<terminated> Question2 (1) [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (23-Feb-2021, 9:21:37 am – 9:21:39 am)

Original-{A=Shreya, B=Riya, C=Rohan, D=Priya}  
Reverse-[D, C, B, A]

3.

The screenshot shows an IDE with a file named `Question3.java` and a console window. The code defines a custom exception `ExceptPrime` and a class `Question3` with a `checkPrime` method and a `main` method. The `main` method iterates from 2 to 100, checking for prime numbers and throwing `ExceptPrime` if a number is not prime. The console output shows the results of the prime checks for each number from 2 to 97.

```

1 package namdson;
2 import java.util.*;
3
4 class ExceptPrime extends Exception{
5
6 }
7
8 public class Question3 {
9     public static boolean checkPrime(int number) {
10         if(number<=1)
11             return false;
12         if(number<=3)
13             return true;
14         if(number%2==0||number%3==0)
15             return false;
16         for(int i=5;i*i<=number;i=i+6)
17             if(number%i==0 || number%(i+2)==0)
18                 return false;
19         return true;
20     }
21
22     public static void main(String[] args) {
23         int c=1;
24         while(c<=100) {
25             try {
26                 if(checkPrime(c))
27                     throw new ExceptPrime();
28             }
29             catch(ExceptPrime e) {
30                 System.out.println(c+" is a Prime Number");
31             }
32             finally {
33                 c++;
34             }
35         }
36     }
37 }

```

Console Output:

```

<terminated> Question3 [Java Applicati
2 is a Prime Number
3 is a Prime Number
5 is a Prime Number
7 is a Prime Number
11 is a Prime Number
13 is a Prime Number
17 is a Prime Number
19 is a Prime Number
23 is a Prime Number
29 is a Prime Number
31 is a Prime Number
37 is a Prime Number
41 is a Prime Number
43 is a Prime Number
47 is a Prime Number
53 is a Prime Number
59 is a Prime Number
61 is a Prime Number
67 is a Prime Number
71 is a Prime Number
73 is a Prime Number
79 is a Prime Number
83 is a Prime Number
89 is a Prime Number
97 is a Prime Number

```

4.

The screenshot shows an IDE with a file named `Question4.java` and a console window. The code defines a `Data` class with transient fields and a `Question4` class with a `display` method and a `main` method. The `main` method creates a `Data` object, writes it to a file, and then prints the state before and after deserialization. The console output shows the state of the object before and after deserialization, demonstrating that transient fields are not serialized.

```

1 Data d = new Data(1, "Snigdha", 22, 2, "CSE");
2
3 transient int id;
4 transient int age;
5 String name;
6 transient int batch;
7 String department;
8
9
10
11
12
13
14
15
16 public Data(int id, String name, int age ,int batch,String department)
17 {
18     this.id = id;
19     this.name = name;
20     this.age = age;
21     this.batch=batch;
22     this.department=department;
23 }
24
25 public class Question4{
26     public static void display(Data d){
27         System.out.println("ID: " + d.id);
28         System.out.println("Name: " + d.name);
29         System.out.println("Age: " + d.age);
30         System.out.println("Batch: " + d.batch);
31         System.out.println("Department: " + d.department);
32     }
33
34     public static void main(String []args){
35         Data d = new Data(1, "Snigdha", 22,2,"CSE");
36         try {
37
38             FileOutputStream file = new FileOutputStream("store.txt");
39             ObjectOutputStream out = new ObjectOutputStream(file);
40             out.writeObject(d);
41             out.close();
42             file.close();
43             System.out.println("Before deserialization:\n ");
44         }
45     }
46 }

```

Console Output:

```

<terminated> Question4 [Java Application] C
Before deserialization:
ID: 1
Name: Snigdha
Age: 22
Batch: 2
Department: CSE
After deserialization:
ID: 0
Name: Snigdha
Age: 0
Batch: 0
Department: CSE

```

## Questions on Slides:

**1.**

```
17 Stack<Integer> a = new Stack<Integer>();
18 Stack<Double> b = new Stack<Double>();
19 Stack<Integer> c = new Stack<Integer>();
20 Stack<Double> d = new Stack<Double>();
21
22 System.out.println("\nEnter string\n");
23 String str = sn.next();
24 str = "0" + str;
25 str = str.replaceAll("-", "+");
26 String temp = "";
27 for (int j = 0; j < str.length(); j++)
28 {
29     char ch = str.charAt(j);
30     if (ch == '-')
31         temp = "-" + temp;
32     else if (ch != '+' && ch != '*' && ch != '/')
33         temp = temp + ch;
34     else
35     {
36         b.push(Double.parseDouble(temp));
37         a.push((int)ch);
38         temp = "";
39     }
40 }
41 b.push(Double.parseDouble(temp));
42 char operators[] = {'/', '*', '+'};
43 for (int i = 0; i < 3; i++)
44 {
45     boolean it = false;
46     while (!a.isEmpty())
47     {
48         int opr = a.pop();
49         double v1 = b.pop();
50         double v2 = b.pop();
51         if (opr == operators[i])
```

**2.**

```
Question2.java
1 age slides;
2 -t java.util.*;
3
4 ic class Question2 {
5
6 public static void main(String[] args) {
7
8     String str1 = "*****\n*****";
9     String str2 = "*****";
10    for (int i = 0; i < 4; i++) {
11        System.out.println(str1);
12    }
13    System.out.println("*****");
14    for (int i = 0; i < 6; i++) {
15        System.out.println(str2);
16    }
17 }
18
```

<terminated> Question2 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin/javaw.exe (23-Feb-2021, 9:27:31 am – 9:27:32 am)