

# Accolite University Feb Batch 1

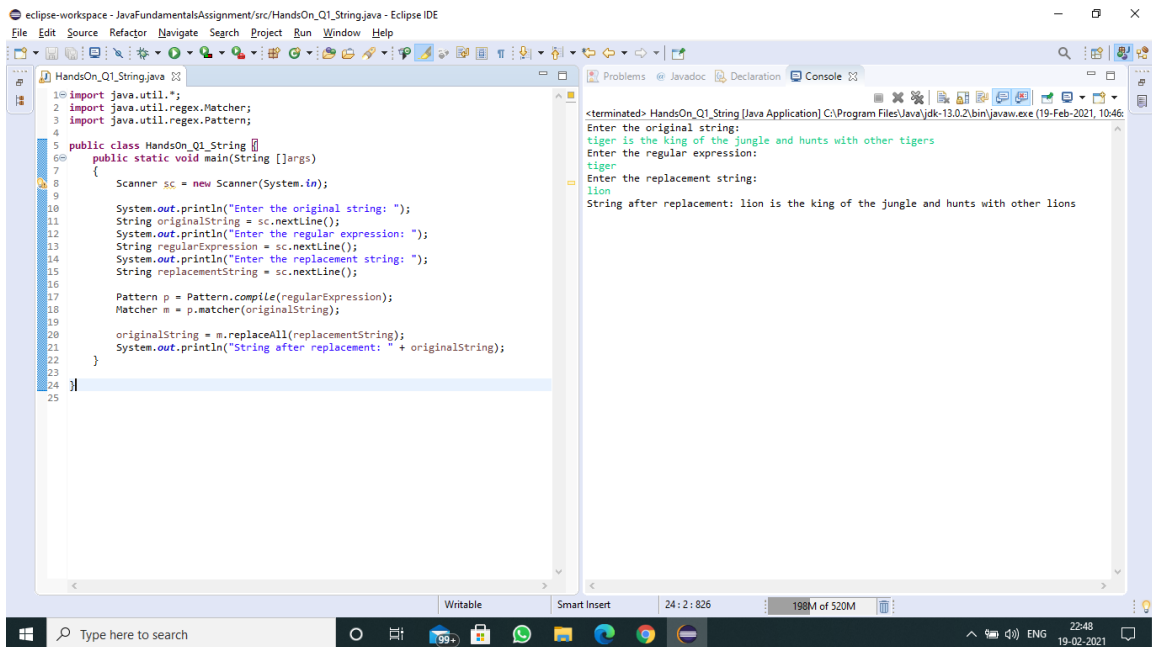
## Java Fundamentals Assignment

- Meet Shah  
(INT626)

ALL SCREENSHOTS

### HandsOn Questions

Q1.



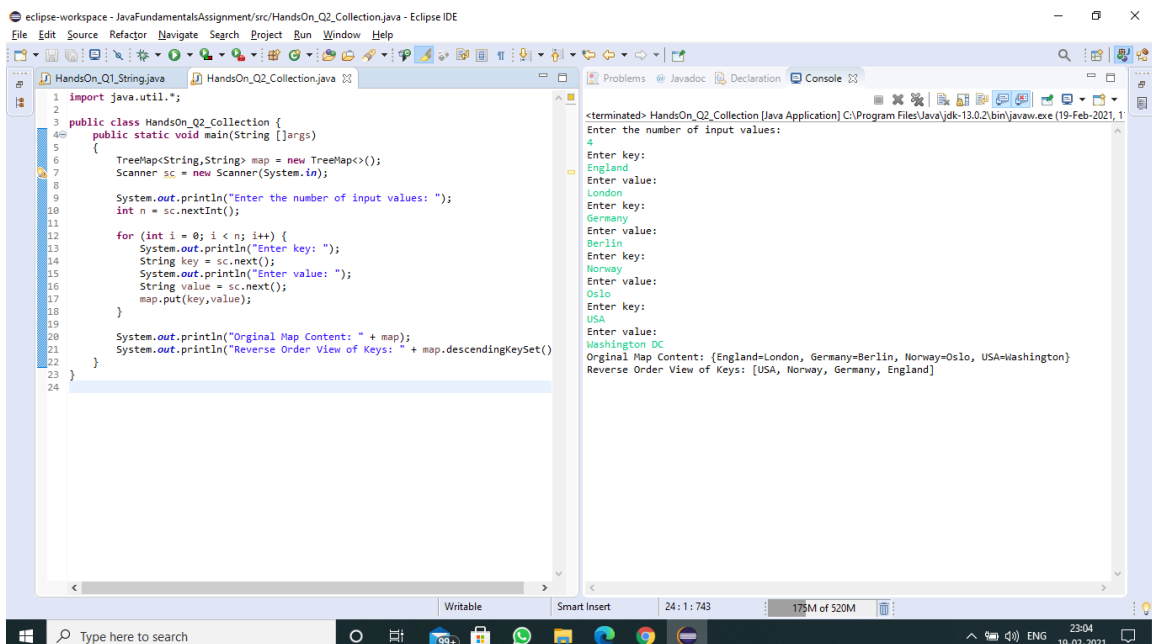
The screenshot shows the Eclipse IDE with a Java project named 'JavaFundamentalsAssignment'. The editor displays the file 'HandsOn\_Q1\_String.java'. The code imports 'java.util.\*', 'java.util.regex.Matcher', and 'java.util.regex.Pattern'. It defines a class 'HandsOn\_Q1\_String' with a 'main' method. The method uses a 'Scanner' to take input from the user. It prompts the user to enter an original string, a regular expression, and a replacement string. It then uses 'Pattern.compile()' and 'Matcher' to replace all occurrences of the regular expression in the original string with the replacement string. The console output shows the program execution: 'Enter the original string: tiger is the king of the jungle and hunts with other tigers', 'Enter the regular expression: tiger', 'Enter the replacement string: lion', and 'String after replacement: lion is the king of the jungle and hunts with other lions'.

```
1 import java.util.*;
2 import java.util.regex.Matcher;
3 import java.util.regex.Pattern;
4
5 public class HandsOn_Q1_String {
6     public static void main(String []args)
7     {
8         Scanner sc = new Scanner(System.in);
9
10        System.out.println("Enter the original string: ");
11        String originalString = sc.nextLine();
12        System.out.println("Enter the regular expression: ");
13        String regularExpression = sc.nextLine();
14        System.out.println("Enter the replacement string: ");
15        String replacementString = sc.nextLine();
16
17        Pattern p = Pattern.compile(regularExpression);
18        Matcher m = p.matcher(originalString);
19
20        originalString = m.replaceAll(replacementString);
21        System.out.println("String after replacement: " + originalString);
22    }
23 }
24
25
```

Console Output:

```
<terminated> HandsOn_Q1_String [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (19-Feb-2021, 10:46)
Enter the original string:
tiger is the king of the jungle and hunts with other tigers
Enter the regular expression:
tiger
Enter the replacement string:
lion
String after replacement: lion is the king of the jungle and hunts with other lions
```

Q2.



The screenshot shows the Eclipse IDE with a Java project named 'JavaFundamentalsAssignment'. The editor displays the file 'HandsOn\_Q2\_Collection.java'. The code imports 'java.util.\*'. It defines a class 'HandsOn\_Q2\_Collection' with a 'main' method. The method uses a 'TreeMap' to store key-value pairs. It prompts the user to enter the number of input values, then enters a loop to take key-value pairs from the user and store them in the 'TreeMap'. After the loop, it prints the 'Original Map Content' and the 'Reverse Order View of Keys'. The console output shows the program execution: 'Enter the number of input values: 4', 'Enter key: England', 'Enter value: London', 'Enter key: Germany', 'Enter value: Berlin', 'Enter key: Norway', 'Enter value: Oslo', 'Enter key: USA', 'Enter value: Washington DC', 'Original Map Content: {England=London, Germany=Berlin, Norway=Oslo, USA=Washington}', and 'Reverse Order View of Keys: [USA, Norway, Germany, England]'.

```
1 import java.util.*;
2
3 public class HandsOn_Q2_Collection {
4     public static void main(String []args)
5     {
6         TreeMap<String,String> map = new TreeMap<>();
7         Scanner sc = new Scanner(System.in);
8
9         System.out.println("Enter the number of input values: ");
10        int n = sc.nextInt();
11
12        for (int i = 0; i < n; i++) {
13            System.out.println("Enter key: ");
14            String key = sc.next();
15            System.out.println("Enter value: ");
16            String value = sc.next();
17            map.put(key,value);
18        }
19
20        System.out.println("Original Map Content: " + map);
21        System.out.println("Reverse Order View of Keys: " + map.descendingKeySet());
22    }
23 }
24
```

Console Output:

```
<terminated> HandsOn_Q2_Collection [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (19-Feb-2021, 11:43)
Enter the number of input values:
4
Enter key:
England
Enter value:
London
Enter key:
Germany
Enter value:
Berlin
Enter key:
Norway
Enter value:
Oslo
Enter key:
USA
Enter value:
Washington DC
Original Map Content: {England=London, Germany=Berlin, Norway=Oslo, USA=Washington}
Reverse Order View of Keys: [USA, Norway, Germany, England]
```

Q3.

The screenshot shows the Eclipse IDE with the file `HandsOn_Q3_Exception.java` open. The code defines a custom exception `PrimeException` and a `main` method that checks for prime numbers from 2 to 100. The console output shows the results of the prime number check.

```

import java.util.*;

class PrimeException extends Exception {
}

public class HandsOn_Q3_Exception {
    public static void main(String []args){
        int count=1;
        while(count<=100) {
            try {
                if(isPrime(count))
                    throw new PrimeException();
            }
            catch(PrimeException e) {
                System.out.println("A prime Number is found:"+count);
            }
            finally {
                count++;
            }
        }
    }

    public static boolean isPrime(int n) {
        if (n <= 1)
            return false;
        for (int i = 2; i < n; i++)
            if (n % i == 0)
                return false;
        return true;
    }
}

```

```

<terminated> HandsOn_Q3_Exception [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (19-Feb-2021, 11:53:09)
A prime Number is found:2
A prime Number is found:3
A prime Number is found:5
A prime Number is found:7
A prime Number is found:11
A prime Number is found:13
A prime Number is found:17
A prime Number is found:19
A prime Number is found:23
A prime Number is found:29
A prime Number is found:31
A prime Number is found:37
A prime Number is found:41
A prime Number is found:43
A prime Number is found:47
A prime Number is found:53
A prime Number is found:59
A prime Number is found:61
A prime Number is found:67
A prime Number is found:71
A prime Number is found:73
A prime Number is found:79
A prime Number is found:83
A prime Number is found:89
A prime Number is found:97

```

Q4.

The screenshot shows the Eclipse IDE with the file `HandsOn_Q4_Serialization.java` open. The code defines an `Employee` class that implements `Serializable` and a `main` method that serializes an `Employee` object to a file and then deserializes it. The console output shows the serialization and deserialization process. A Notepad window is open, displaying the raw bytes of the serialized object.

```

import java.io.*;

class Employee implements Serializable {
    UUID serialversionUID = new UUID(100,10);
    transient int id;
    String name;
    int age;
    int salary;
    transient int experience;

    public Employee(int id, String name, int age, int salary, int experience) {
        this.id = id;
        this.name = name;
        this.age = age;
        this.salary = salary;
        this.experience = experience;
    }
}

public class HandsOn_Q4_Serialization {
    public static void main(String[] args){
        Employee emp = new Employee(12345, "Meet", 21, 1100000, 1);
        String filename = "file.txt";
        try {
            FileOutputStream file = new FileOutputStream(filename);
            ObjectOutputStream out = new ObjectOutputStream(file);
            out.writeObject(emp);
            out.close();
            file.close();
            System.out.println("Object serialized....");
            System.out.println("Data before deserialization: ");
            display(emp);
        }
        catch (Exception e) {}
        emp = null;
        System.out.println("-----");
        try {
            FileInputStream file = new FileInputStream(filename);

```

```

<terminated> HandsOn_Q4_Serialization [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (19-Feb-2021, 11:53:09)
Object serialized....
Data before deserialization:
id = 12345
name = Meet
age = 21
salary = 1100000
experience = 1
-----
Object deserialized....
Data after deserialization:
id = 0
name = Meet
age = 21
salary = 1100000
experience = 0

```

```

File - Notepad
File Edit Format View Help
-i [sr] [Employeeid] [age] [salary] [name] [Ljava/lang/String;L
[serialversionUID] [Ljava/util/UUID;xp [0] [0] [Meet]
[Ljava.util.UUID; [m.../ [leastSigBits] [mostSigBitsxp
d

```

Q1.

Q2.

[illegible]