

1. Write a java program

i. to compare two strings lexicographically, ignoring case differences.

```
import java.util.*; import java.Text.*;
```

```
public class StringOperations {
```

```
    public static void main(String[] args) {
```

i. Compare two strings lexicographically, ignoring case differences. String str1 = "Hello";

```
    String str2 = "hELLO";
```

```
    int result = str1.compareToIgnoreCase(str2);
```

```
        if (result == 0) {
```

```
            System.out.println("Strings are equal.");
```

```
        } else if (result < 0) {
```

```
            System.out.println("String 1 is lexicographically smaller than String 2.");
```

```
        } else {
```

```
            System.out.println("String 2 is lexicographically smaller than String 1.");
```

```
        }
```

ii. Check whether a given string ends with the contents of another string.

```
    String mainStr = "Hello World";      String
```

```
    suffixStr = "World";
```

```
    boolean endsWith = mainStr.endsWith(suffixStr);
```

```
        if (endsWith) {
```

```
            System.out.println("Main string ends with the given suffix string.");
```

```
        }
```

```
    else {
```

```
        System.out.println("Main string does not end with the given suffix string.");
```

```
    }
```

iii. Print current date and time in the specified format.

```
Date date = new Date();  
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");  
String formattedDate = sdf.format(date);  
System.out.println("Current date and time: " + formattedDate);
```

iv. Get the index of all the characters of the alphabet. String str =

```
"abcdefghijklmnopqrstuvwxyz";      for (char ch = 'a'; ch <= 'z'; ch++) {  
int index = str.indexOf(ch);  
        System.out.println("Index of " + ch + ": " + index);  
    }
```

v. Replace each substring of a given string that matches the given regular expression with the given replacement.

```
String inputStr = "The quick brown fox jumps over the lazy dog. The quick brown fox jumps over  
the lazy dog.";  
String regexStr = "fox";  
String replacementStr = "cat";  
String outputStr = inputStr.replaceAll(regexStr, replacementStr);  
System.out.println("Output string: " + outputStr);
```

vi. Get a substring of a given string between two specified positions. String input = "Hello

```
World";      int startIndex = 1;      int endIndex = 6;  
String output = input.substring(startIndex, endIndex);  
System.out.println("Substring: " + output);
```

vii. Trim any leading or trailing whitespace from a given string.

```
String strToTrim = " Hello World  ";  
String trimmedStr = strToTrim.trim();
```

```
System.out.println("Trimmed string: " + trimmedStr);  viii.
```

Convert all the characters in a string to lowercase.

```
String inputString = "Hello World";
```

```
String outputString = inputString.toLowerCase();
```

```
System.out.println("Output string: " + outputString);
```

ix. Get the length of a given string. `String lenStr = "Hello`

```
World";    int length = lenStr.length();
```

```
System.out.println("Length of the string: " + length);
```

x. Check whether two String objects contain the same data.

```
String strA = "Hello World";    String
```

```
strB = "Hello World";    boolean
```

```
areEqual = strA.equals(strB);
```

```
if (areEqual) {
```

```
    System.out.println("The two strings contain the same data.");
```

```
} else {
```

```
    System.out.println("The two strings do not contain the same data.");
```

```
}
```

```
}
```

```
}
```

2. Implement a class Account. An account has

- a balance
- functions to add
- and withdraw money,
- And a function to inquire the current balance.

```
public class Account
{
    private double balance; public void
    Account(double initialBalance) {
        this.balance = initialBalance;
    }
    public void Account() {
        this.balance = 0;
    }
    public void addMoney(double amount) {
        this.balance += amount;
    }
    public void withdrawMoney(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient funds. A $5 penalty will be charged.");
            this.balance -= 5;
        } else {
            this.balance -= amount;
        }
    }
}

public double getCurrentBalance() {
    return balance;
}

public double computeInterest(double interestRate) {
    double interest = balance * interestRate / 100;
    this.balance += interest;    return interest;
}
```

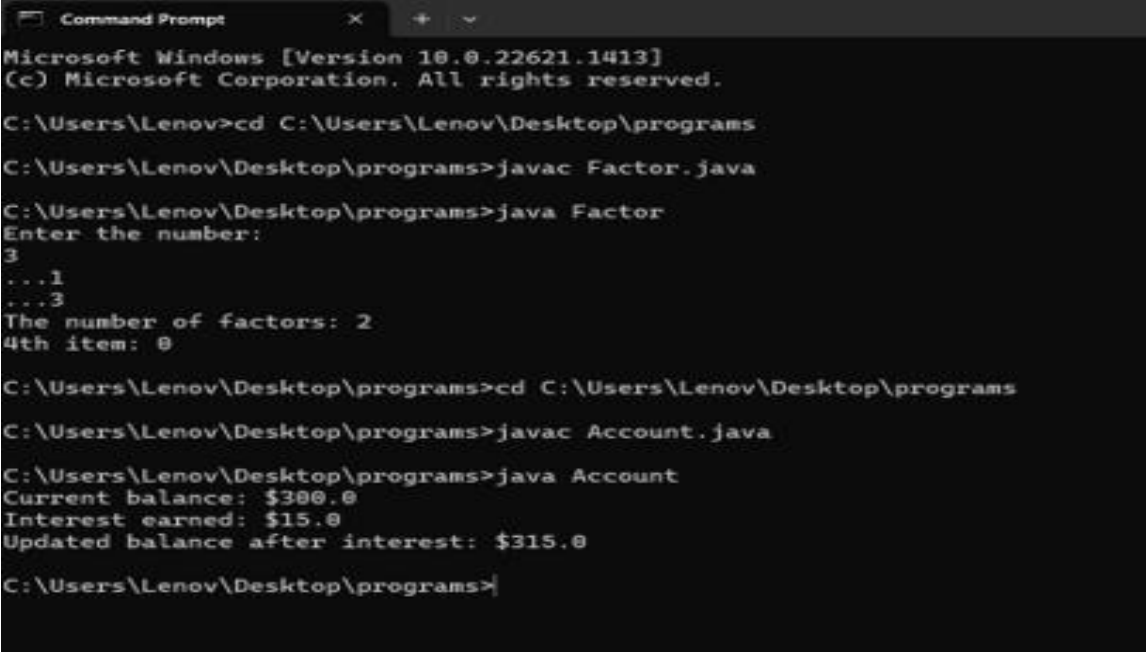
```

public static void main(String[] args) {
    Account myAccount = new Account();
    myAccount.addMoney(500);
    myAccount.withdrawMoney(200);

    double balance = myAccount.getCurrentBalance();
    System.out.println("Current balance: $" + balance);
    double interest = myAccount.computeInterest(5);
    System.out.println("Interest earned: $" + interest);
    System.out.println("Updated balance after interest: $" + myAccount.getCurrentBalance());
}
}

```

Output:



```

Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenov>cd C:\Users\Lenov\Desktop\programs
C:\Users\Lenov\Desktop\programs>javac Factor.java
C:\Users\Lenov\Desktop\programs>java Factor
Enter the number:
3
...1
...3
The number of factors: 2
4th item: 0

C:\Users\Lenov\Desktop\programs>cd C:\Users\Lenov\Desktop\programs
C:\Users\Lenov\Desktop\programs>javac Account.java
C:\Users\Lenov\Desktop\programs>java Account
Current balance: $300.0
Interest earned: $15.0
Updated balance after interest: $315.0
C:\Users\Lenov\Desktop\programs>

```

3. Given two strings needle and haystack, return the index of the first occurrence of needle in haystack, or -1 if needle is not part of haystack.

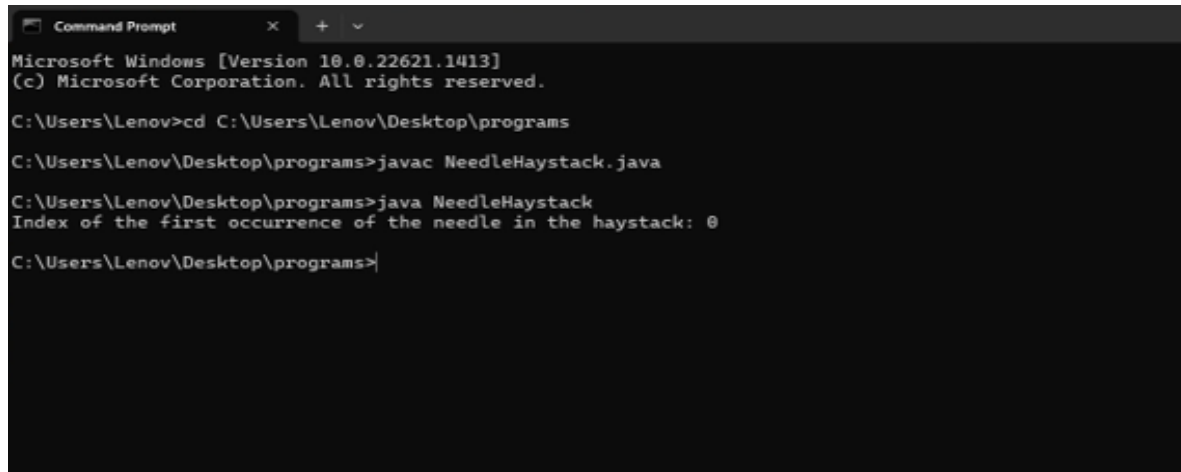
Program:

```
public class NeedleHaystack {

    public static int findNeedle(String haystack, String needle) {
        int n = haystack.length();    int m = needle.length();
        if (m == 0) {
            return 0;
        }
        for (int i = 0; i <= n - m; i++) {    if
(haystack.substring(i, i + m).equals(needle)) {
            return i;
        }
    }
    return -1;
}

    public static void main(String[] args) {
        String haystack = "sadbutsad";
        String needle = "sad";
        int index = findNeedle(haystack, needle);
        System.out.println("Index of the first occurrence of the needle in the haystack: " + index);
    }
}
```

Output:



```
Command Prompt
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenov>cd C:\Users\Lenov\Desktop\programs

C:\Users\Lenov\Desktop\programs>javac NeedleHaystack.java

C:\Users\Lenov\Desktop\programs>java NeedleHaystack
Index of the first occurrence of the needle in the haystack: 0

C:\Users\Lenov\Desktop\programs>|
```

4. Questions for Finding error in Java to determine the factor.

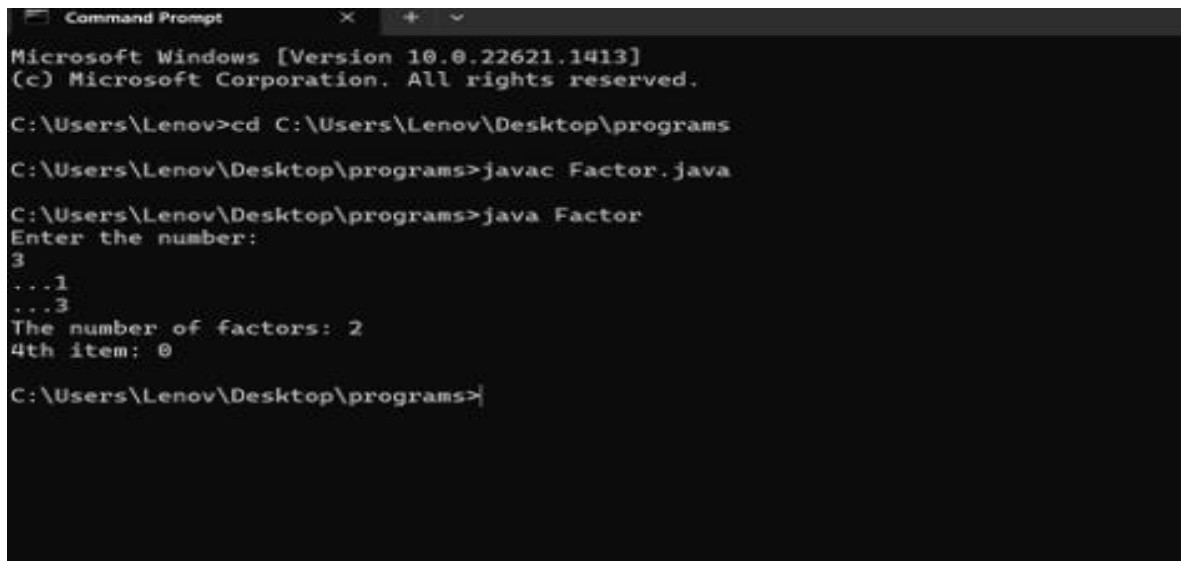
```
import java.util.*; class Factor { public static void
main(String args[]) {
    try {
        Scanner sc = new Scanner(System.in);      int
count = 0, n = 100, i, j = 0, m = 4;      int[]
a = new int[10];
        System.out.println("Enter the number:");
        n = sc.nextInt();      if
(n <= 0) {
            System.out.println("Enter valid number");
        } else {      for (i =
1; i <= n; i++) {      if (n %
i == 0) {      a[j] =
i;
            System.out.println("..." + i);      count++;
j++;
        }
    }
}
```

```

    }
    System.out.println("The number of factors: " + count);
}
System.out.println(m + "th item: " + a[m - 1]);
} catch (Exception e) {
    System.out.println("Enter only numbers");
}
}
}

```

Out put:



```

Command Prompt
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenov>cd C:\Users\Lenov\Desktop\programs
C:\Users\Lenov\Desktop\programs>javac Factor.java
C:\Users\Lenov\Desktop\programs>java Factor
Enter the number:
3
...1
...3
The number of factors: 2
4th item: 0
C:\Users\Lenov\Desktop\programs>

```