Name: **Shangirne Kharbanda**

Registration Number: **20BAI1154**

JAVA LAB-1

STRINGS

1.Write an application that prompts the user for three first names and concatenates them in every possible two-name combination so that new parents can easily compare them to find the most pleasing baby name. Save the file as BabyNameComparison.java.

BabyNameComparison.java

```
import java.util.Scanner;

public class BabyNameComparison {
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        String[] name  = new String[3];
        for(int i=0;i<3;i++){
            name[i] = sc.nextLine();
        }
        for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                System.out.println(name[i] +" "+ name[j]);
            }
        }

    }
}
```

Output 1:

```
Mary
John
Daffodil
Mary Mary
Mary John
Mary Daffodil
John Mary
John John
John Daffodil
Daffodil Mary
Daffodil John
Daffodil Daffodil

Process finished with exit code 0
```

2. a. Create a program that contains a String that holds your favorite inspirational quote and display the total number of spaces contained in the String. Save the file as CountSpaces.java. b. Write an application that counts the total number of spaces contained in a quote entered by the user. Save the file as CountSpaces2.java.

CountSpaces.java

```
public class CountSpaces {
    public static void main(String args[]){
            String quote = "Rise and shine and get your work done!";
            int n = 0;
            for(int i=0;i<quote.length();i++){
                char ch = quote.charAt(i);
                if(ch==' '){
                    n++;
                }
            }
            System.out.println("The number of spaces in the quote is " +
n);
    }
}
```

Output:

```
The number of spaces in the quote is 7


Process finished with exit code 0
```

CountSpaces2.java

```
import java.util.Scanner;
public class CountSpaces2 {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your quote:");
        String quote = sc.nextLine();
        int n = 0;
        for (int i = 0; i < quote.length(); i++) {
            char ch = quote.charAt(i);
            if (ch == ' ') {
                n++;
            }
        }
        System.out.println("The number of spaces in the quote is " + n);
    }
}
```

Output:

```
Enter your quote:
Do your work and spend your life working all the time
The number of spaces in the quote is 10

Process finished with exit code 0
```

3. Write an application that prompts the user for a password that contains at least two uppercase letters, at least three lowercase letters, and at least one digit. Continuously reprompt the user until a valid password is entered. Display a message indicating whether the password is valid; if not, display the reason the password is not valid.

Password.java

```
import java.util.Scanner;

public class Password {
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        while(true) {
            int u = 0, l = 0, d = 0;
            System.out.println("Enter password:");
            String password = sc.nextLine();
            for (int i = 0; i < password.length(); i++) {
                char ch = password.charAt(i);
                if (Character.isUpperCase(ch)) {
                    u++;
                }
                if (Character.isLowerCase(ch)) {
                    l++;
                }
                if (Character.isDigit(ch)) {
                    d++;
                }
            }

            if (u >= 2 && l >= 3 && d >= 1) {
                System.out.println("Your password is valid.");
                break;
            } else {
                System.out.println("Your password is invalid. Please try
again.");
            }
        }


    }
}
```

Output:

```
Enter password:
Iadad
Your password is invalid. Please try again.
Enter password:
IDkwfa2
Your password is valid.

Process finished with exit code 0
```

4. Write an application that counts the words in a String entered by a user. Words are separated by any combination of spaces, periods, commas, semicolons, question marks, exclamation points, or dashes. Figure 7-17 shows a typical execution. Save the file as CountWords.java.

CountWords.java

```
import java.util.Scanner;

public class CountWords {
    static int wordcount(String string)
    {
        int count=0;

        char ch[] = new char[string.length()];
        for(int i=0;i<string.length();i++)
        {
            ch[i]= string.charAt(i);
            if( ((i>0)&&(ch[i]!=' ')&&(ch[i-1]==' ')) || ((ch[0]!='
')&&(i==0)) )
                count++;
        }
        return count;
    }
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your string:");
        String str = sc.nextLine();
        System.out.println(wordcount(str) + " words");


    }
}
```

Output:

```
Enter your string:
I am doing     greattt
4 words

Process finished with exit code 0
```

5. Write an application that accepts three Strings from the user and displays them in alphabetical order without regard to case. Save the file as Alphabetize2.java.

Alphabetize2.java

```
import java.util.Scanner;
public class Alphabetize2{

    public static void main(String []args){
        String temp;
        Scanner sc=new Scanner(System.in);

        System.out.println("Enter 1st string");
        String s1=sc.next();
        System.out.println("Enter 2nd string");
        String s2=sc.next();
        System.out.println("Enter 3rd string");
        String s3=sc.next();

        if (s1.compareTo(s2)>0)
        {
            temp =s1;
            s1 = s2;
            s2 = temp;
        }
        if (s2.compareTo(s3)>0)
        {
            temp =s2;
            s2 = s3;
            s3 = temp;
        }
        if (s1.compareTo(s3)>0)
        {
            temp =s1;
            s1 = s3;
            s3 = temp;
        }
        System.out.print("strings in Sorted Order:");
        System.out.println(s1);
        System.out.println(s2);
        System.out.println(s3);
    }
}
```

Output:

```
Enter 1st string
hello
Enter 2nd string
bye
Enter 3rd string
hola
strings in Sorted Order:bye
hello
hola

Process finished with exit code 0
```

6. Write an application that accepts a word from a user and converts it to Pig Latin. If a word starts with a consonant, the Pig Latin version removes all consonants from the beginning of the word and places them at the end, followed by ay. For example, cricket becomes icketcray. If a word starts with a vowel, the Pig Latin version is the original word with ay added to the end. For example, apple becomes appleay. If y is the first letter in a word, it is treated as a consonant; otherwise, it is treated as a vowel. For example, young becomes oungyay, but system becomes ystemsay. For this program, assume that the user will enter only a single word consisting of all lowercase letters. Save the file as PigLatin.java.

PigLatin.java

```
import java.util.Scanner;

public class PigLatin {
    public static String toString(char[] str)
    {
        String string = new String(str);
        return string;
    }
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your word:");
        String word = sc.nextLine();
        char ch[] = new char[word.length()];
        char ch1[] = new char[word.length()];
        char ch2[] = new char[word.length()];
        int j = 0, z = 0,x=0,u=0;
        for(int i=0;i<word.length();i++){
            ch[i] = word.charAt(i);
        }
        for(int i=0;i<word.length();i++) {
            if(ch[i]=='y' && i!=0){
                break;
            }
            if (ch[i] != 'a' && ch[i] != 'e' && ch[i] != 'i' && ch[i] !=
'o' && ch[i] != 'u') {
                ch1[j] = ch[i];
                j++;
```

```
                ch[i] = ' ';
            }
            if(ch[i] == 'a' || ch[i] == 'e' || ch[i] == 'i' || ch[i] == 'o'
|| ch[i] == 'u' || (ch[i]=='y' && i!=0)){
                break;
            }
        }
        for(int i=0;i<word.length();i++){
            if(ch[i]!=' '){
                ch2[x]=ch[i];
                x++;
            }

        }
        for(int i=0;i<word.length();i++){
            if(i>=x){
                ch2[i]=ch1[u];
                u++;
            }

        }

        System.out.println("Pig Latin version of entered word:");
        System.out.println(toString(ch2) + "ay");

    }
}
```

Output:

```
Enter your word:
cricket
Pig Latin version of entered word:
icketcray

Process finished with exit code 0
```

```
Enter your word:
system
Pig Latin version of entered word:
ystemsay

Process finished with exit code 0
```

7. Write a program that inserts parentheses, a space, and a dash into a string of 10 user-entered numbers to format it as a phone number. For example, 5153458912 becomes (515) 345-8912. If the

user does not enter exactly 10 digits, display an error message. Continue to accept user input until the user enters 999.

Mobile.java

```
import java.util.Scanner;

public class Mobile {
    public static String toString(char[] str)
    {
        String string = new String(str);
        return string;
    }
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        while(true){
            System.out.println("Enter your phone number:");
            String ph = sc.nextLine();
            if(ph.equals("999")){
                System.out.println("Thank you!");
                break;
            }
            if(ph.length()!=10){
                System.out.println("Error! Try again.");
                continue;
            }


            char phone[] = new char[ph.length()];
            char phone1[] = new char[ph.length()+4];
            int u=0;
            for(int i=0;i<ph.length();i++){
                phone[i]=ph.charAt(i);
            }
            for(int i=0;i<ph.length()+4;i++){
                if(i==0){
                    phone1[i] = '(';
                    continue;
                }
                if(i==4){
                    phone1[i]= ')';
                    continue;
                }
                if(i==5){
                    phone1[i]=' ';
                    continue;
                }
                if(i==9){
                    phone1[i]='-';
                    continue;
                }
                phone1[i]=phone[u];
                u++;

            }

            System.out.println("The phone number format of entered phone
is: " + toString(phone1));
```

```
        }


    }
}
```

Output:

```
Enter your phone number:
9976587654
The phone number format of entered phone is: (997) 658-7654
Enter your phone number:
1234
Error! Try again.
Enter your phone number:
999
Thank you!

Process finished with exit code 0
```

8. Write an application that determines whether a phrase entered by the user is a palindrome. A palindrome is a phrase that reads the same backward and forward without regarding capitalization or punctuation. For example, “Dot saw I was Tod”, “Was it a car or a cat I saw?”, and “Madam, I’m Adam” are palindromes. Save the file as Palindrome.java.

Palindrome.java

```
import java.util.Scanner;

public class Palindrome {
        static boolean stringPalindrome(String str)
        {
            int l = 0;
            int h = str.length()-1;
            str = str.toLowerCase();
            while(l <= h)
            {
                char getAtl = str.charAt(l);
                char getAth = str.charAt(h);
                if (!(getAtl >= 'a' && getAtl <= 'z'))
                    l++;
                else if(!(getAth >= 'a' && getAth <= 'z'))
                    h--;
                else if( getAtl == getAth)
                {
                    l++;
                    h--;
```

```
                }
                else
                    return false;
            }
            return true;
        }

        public static void main(String[] args)
        {
            Scanner sc = new Scanner(System.in);
            String str = sc.nextLine();
            if( stringPalindrome(str))
                System.out.println("String is palindrome");
            else
                System.out.println("String is not" + " " + "palindrome");
        }
}
```

Output:

```
Dot saw i was tod
String is palindrome

Process finished with exit code 0
```

```
Evil olive
String is palindrome

Process finished with exit code 0
```