OBJECT ORIENTED PROGRAMMING WITH JAVA 8– LAB 9

Q1. Write a program to find if a given string is a palindrome or not using StringBuilder.

```
import java.util.Scanner;
class PalindromeStringBuilder
{
  public static void main(String args[])
  {
    String s1 = "radar";
    StringBuilder sb = new StringBuilder(s1);
    String s2 = sb.reverse().toString();
    System.out.println(s2);
    if(s1.equals(s2))
        System.out.println("String is Palindrome");
    else
        System.out.println("String is not a Palindrome");
    }
}
```

```
E:\java_notes>javac PalindromeStringBuilder.java

E:\java_notes>java PalindromeStringBuilder
radar
String is Palindrome
```

Q2. Accept a line of text. Find the reverse of each word and display the string.

```
import java.util.Scanner;
class ResverseWord
{
 public static String reverseWords(String str)
 {
   String [] words = str.split(" ");
   StringBuilder rt = new StringBuilder();
    for (String word : words)
           {
            StringBuilder rb = new StringBuilder(word);
            String w = rb.reverse().toString();
            rt.append(w).append(" ");
           }
              return rt.toString().trim();
  }
  public static void main(String[] args)
  {
   Scanner sc = new Scanner(System.in);

   System.out.print("Enter a line of text: ");
   String input = sc.nextLine();

   String rt = reverseWords(input);

   System.out.println("Reversed text: " + rt);
}}
```

```
E:\java_notes>javac ResverseWord.java

E:\java_notes>java ResverseWord
Enter a line of text: good morning
Reversed text: doog gninrom
```

Q3. A sentence is terminated by either ".", "!", "?". Input a piece of text containing sentences. Obtain the length of the sentence and frequency of vowels in each sentence.

```
class SentenceCheck
{
  public static void main(String args[])
  {
    String text = "This is PG-DBDA course.Has many modules ! is java easy?";
    String [] S = text.split("[.!?]");
    for(String sentence : S) {
        StringBuilder sb = new StringBuilder(sentence);
        int sl = sb.length();
        int vowelCount = 0;
        sb = new StringBuilder(sb.toString().toLowerCase());
     for (int i = 0; i <sb.length(); i++) {
        char c = sb.charAt(i);
        if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
        vowelCount++;
        }
       }
     System.out.println("Sentence : " + sentence.trim());
     System.out.println("Length : " + sl);
     System.out.println("Vowel count : " + vowelCount );
     System.out.println();
   }
}
}
```

```
E:\java_notes>javac SentenceCheck.java

E:\java_notes>java SentenceCheck
Sentence : This is PG-DBDA course
Length : 22
Vowel count : 6

Sentence : Has many modules
Length : 17
Vowel count : 5

Sentence : is java easy
Length : 13
Vowel count : 5
```

Q4Create a functional interface named Verify with one abstract method boolean check(int a). Create a class that implements this interface using lambda expression where the method returns true if the number is prime and false otherwise.

```
@FunctionalInterface
interface Verify
{
  public boolean check (int a);
}
 class checkPrime
{
  public static void main(String[] args)
  {
    Verify x = (int a) -> {
      if (a<=1) {
          return false;
      }
      if (a<=3) {
          return true;
      }
      if (a%2 == 0 || a%3==0) {
          return false;
      }
    for (int i = 5; i * i <= a; i += 7) {
       if (a % i == 0 || a % (i + 2) == 0) {
          return false;
        }
      }
          return true;
     };

       int n = 19;
       boolean xresult = x.check(n);
        if (xresult)
           System.out.println(n + " is prime");
        else
           System.out.println(n + "is not prime");

       }
 }
```

```
E:\java_notes>javac checkPrime.java

E:\java_notes>java checkPrime
19 is prime
```

Q5. 5. Create a functional interface named Concatenation with one abstract method String join(String a, String b). Write a program to implement lambda expression to concatenate two strings.

```
@FunctionalInterface
interface Concatenation
{
  String join(String a,String b);
}
class concatString
{
  public static void main(String[] args)
  {
    Concatenation conc = (String a,String b) -> a + b;
    String s1 = " Good Morning ";
    String s2 = "Everyone";
    String x = conc.join(s1,s2);
    System.out.println(x);
  }
}
```

```
E:\java_notes>javac concatString.java

E:\java_notes>java concatString
 Good Morning Everyone
```