Programming Paradigms Laboratory

B.Tech.

Name : KAUSHAL VASHISTH

Roll Number : 18ETCS002147

Department : Computer Science and Engineering

Faculty of Engineering & Technology

Ramaiah University of Applied Sciences

Faculty	Engineering & Technology
Programme	B. Tech. in Computer Science and Engineering
Year/Semester	2nd Year / 4th Semester
Name of the Laboratory	Programming Paradigms Laboratory
Laboratory Code	19CSL217A

Laboratory 5

Title of the Laboratory Exercise: Strings

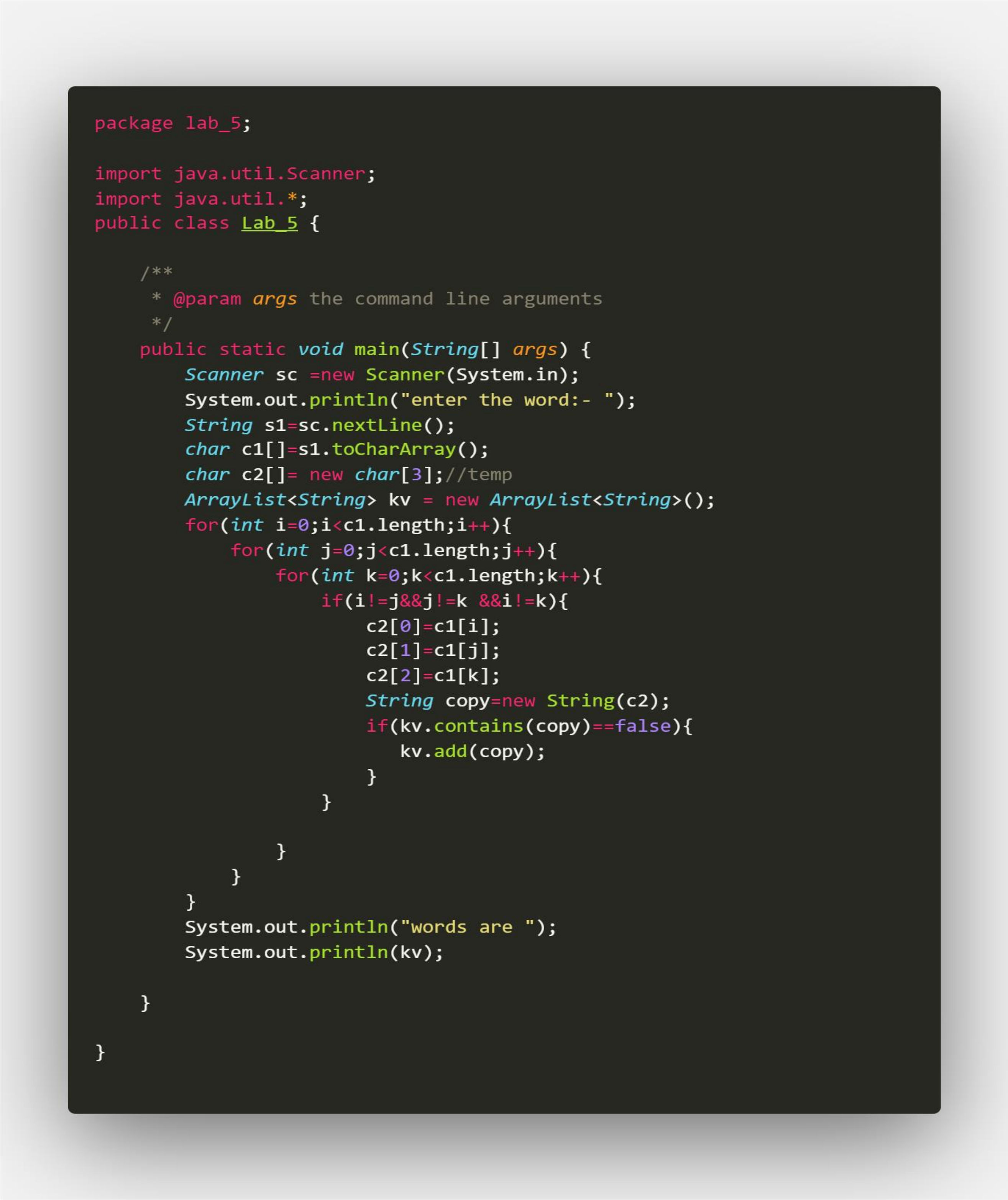
1. Questions

   a. Develop a java method to implement the similar functionality of compareTo() method of String class.

   b. Develop a Java program to reads a five-letter word from the user and produces every possible three-letter string that can be derived from the letters of that word. For example, the three-letter words produced from the word “bathe” include “ate,” “bat,” “bet,” “tab,” “hat,” “the” and “tea.”

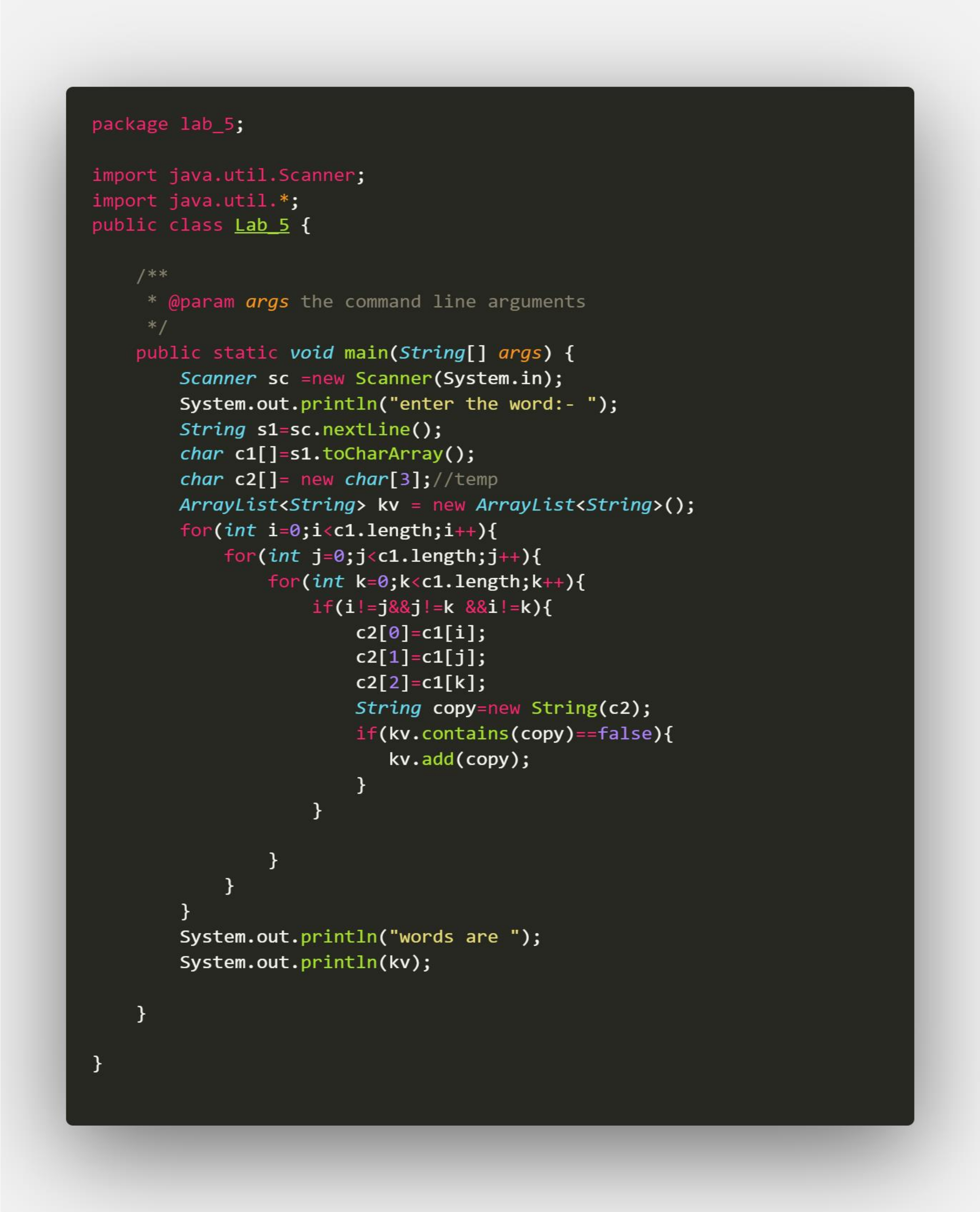
2. Calculations/Computations/Algorithms:-

Part A:-



```
package lab_5;

import java.util.Scanner;
import java.util.*;
public class Lab_5 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        System.out.println("enter the word:- ");
        String s1=sc.nextLine();
        char c1[]=s1.toCharArray();
        char c2[]= new char[3];//temp
        ArrayList<String> kv = new ArrayList<String>();
        for(int i=0;i<c1.length;i++){
            for(int j=0;j<c1.length;j++){
                for(int k=0;k<c1.length;k++){
                    if(i!=j&&j!=k &&i!=k){
                        c2[0]=c1[i];
                        c2[1]=c1[j];
                        c2[2]=c1[k];
                        String copy=new String(c2);
                        if(kv.contains(copy)==false){
                           kv.add(copy);
                        }
                    }

                }
            }
        }
        System.out.println("words are ");
        System.out.println(kv);

    }

}
```

Part b:-

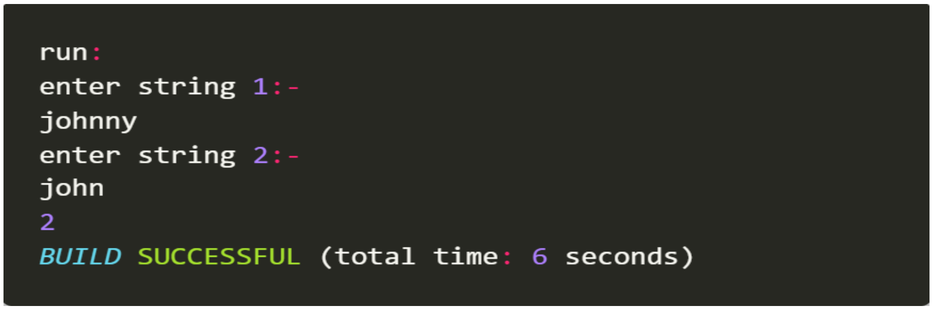


```
package lab_5;

import java.util.Scanner;
import java.util.*;
public class Lab_5 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        System.out.println("enter the word:- ");
        String s1=sc.nextLine();
        char c1[]=s1.toCharArray();
        char c2[]= new char[3];//temp
        ArrayList<String> kv = new ArrayList<String>();
        for(int i=0;i<c1.length;i++){
            for(int j=0;j<c1.length;j++){
                for(int k=0;k<c1.length;k++){
                    if(i!=j&&j!=k &&i!=k){
                        c2[0]=c1[i];
                        c2[1]=c1[j];
                        c2[2]=c1[k];
                        String copy=new String(c2);
                        if(kv.contains(copy)==false){
                           kv.add(copy);
                        }
                    }

                }
            }
        }
        System.out.println("words are ");
        System.out.println(kv);

    }

}
```

3. Presentation of Results:

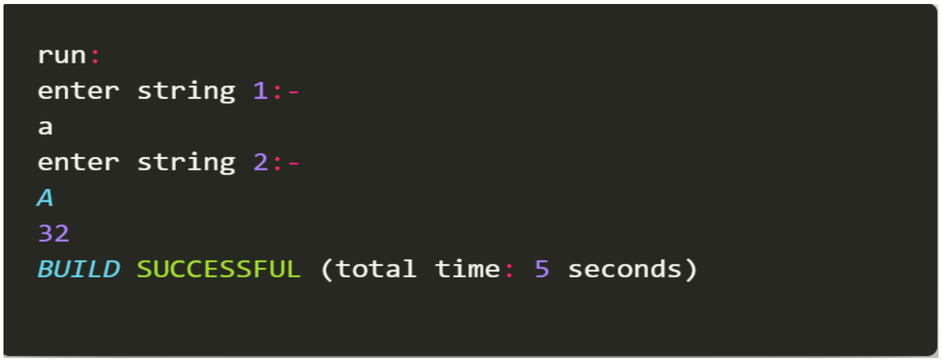
Part A:-

Test case 1:-



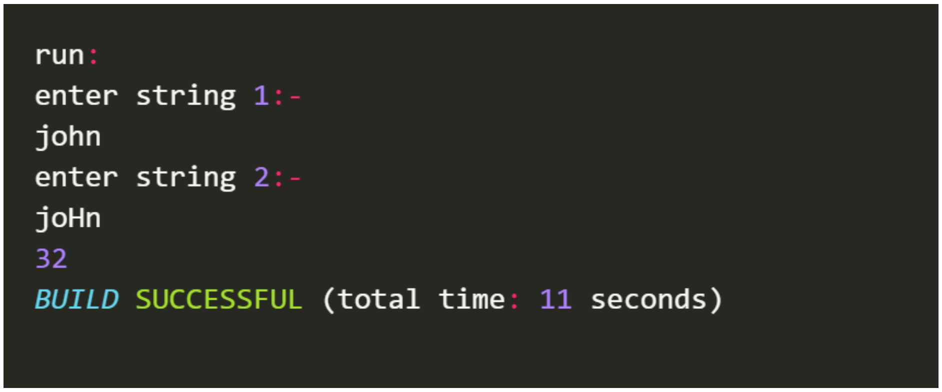
```
run:
enter string 1:-
johnny
enter string 2:-
john
2
BUILD SUCCESSFUL (total time: 6 seconds)
```

Test Case 2:-



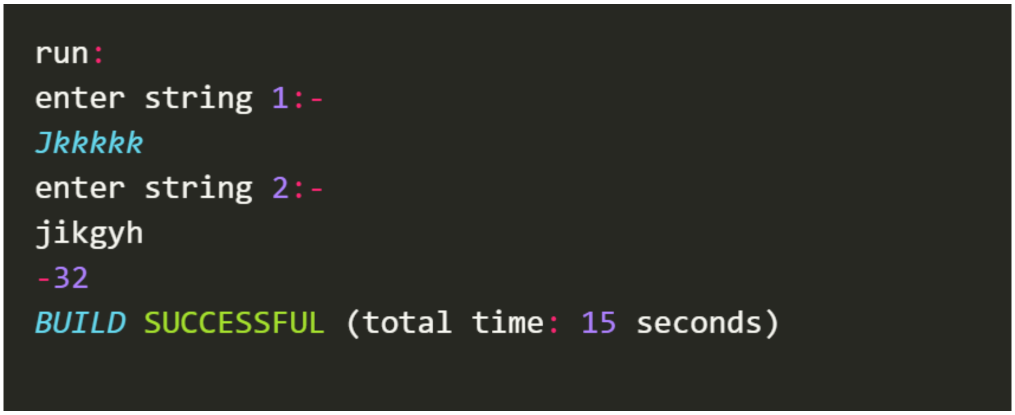
```
run:
enter string 1:-
a
enter string 2:-
A
32
BUILD SUCCESSFUL (total time: 5 seconds)
```

Test Case 3:-



```
run:
enter string 1:-
john
enter string 2:-
joHn
32
BUILD SUCCESSFUL (total time: 11 seconds)
```

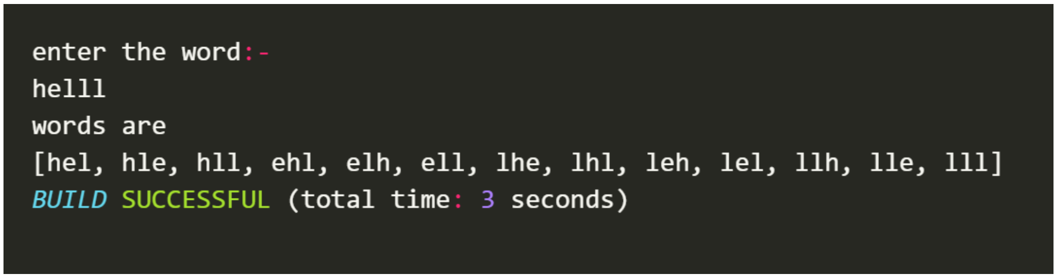
Test Case 4:-



```
run:
enter string 1:-
Jkkkkk
enter string 2:-
jikgyh
-32
BUILD SUCCESSFUL (total time: 15 seconds)
```

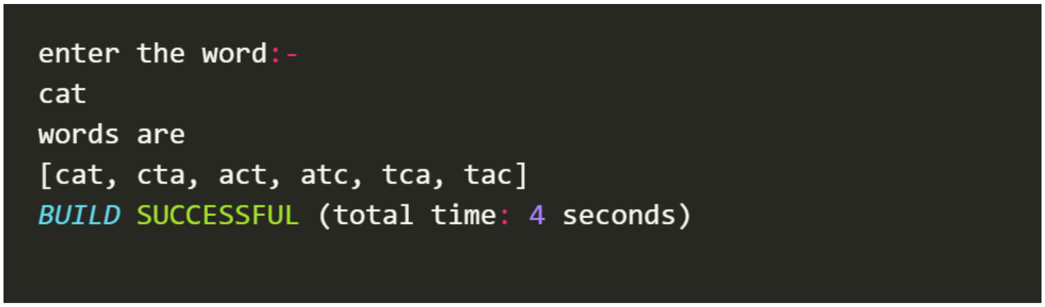
Part B:-

Test case 1:-



```
enter the word:-
helll
words are
[hel, hle, hll, ehl, elh, ell, lhe, lhl, leh, lel, llh, lle, lll]
BUILD SUCCESSFUL (total time: 3 seconds)
```

Test case 2:-



```
enter the word:-
cat
words are
[cat, cta, act, atc, tca, tac]
BUILD SUCCESSFUL (total time: 4 seconds)
```

4. Conclusions :-

   Successfully executed programs about strings in java.

5. Limitations of Experiments and Results

   Disadvantages of using strings in java:

   - Not being able to subclass String means that it is not possible to add behavior to String for your own needs.
   - The previous point means that all access must be through the restricted set of currently available String methods, imposing extra overhead.
   - The only way to increase the number of methods allowing efficient manipulation of String characters is to copy the characters into your own array and manipulate them directly, in which case String is imposing an extra step and extra objects you may not need.
   - char arrays are faster to process directly.