Name: Phạm Duy Đạt ID: DE140191 Class: SE1403

**Question:**

**1. What is the initial capacity of the following?**  
**StringBuilder sb = new StringBuilder("Able was I ere I saw Elba.");**

**Answers:**

Default capacity of StringBuilder is 16

The initial capacity of the above string builder is sb.length() + 16 = 26 + 16 = 42

**2. Consider the following string:  
String hannah = "Did Hannah see bees? Hannah did.";**

**a. What is the value displayed by the expression hannah.length()?  
b. What is the value returned by the method call hannah.charAt(12)?  
c. Write an expression that refers to the letter b in the string referred to by hannah.**

1. hannah.length = 32.
2. hannah.chatAt(12) = ‘e’.
3. hannah.charAt(15)

**3. How long is the string returned by the following expression? What is the string?  
"Was it a car or a cat I saw?".substring(9, 12)**

**Answers:**

It's 3 characters in length. Because SubString(9,12) will take the character at the index number 9 to index number 11

So the String return is “car” with no space  
**4. In the following program, called ComputeResult, what is the value of result after each numbered** **line executes?**

public class ComputeResult {

    public static void main(String[] args) {

        String original = "software";

        StringBuilder result = new StringBuilder("hi");

        int index = original.indexOf('a');

/\*1\*/   result.setCharAt(0, original.charAt(0));

/\*2\*/   result.setCharAt(1, original.charAt(original.length()-1));

/\*3\*/   result.insert(1, original.charAt(4));

/\*4\*/   result.append(original.substring(1,4));

/\*5\*/   result.insert(3, (original.substring(index, index+2) + " "));

        System.out.println(result);

    }

}

**Answer:**

1. si
2. se
3. swe
4. sweoft
5. swear oft

**Exercise:**

**1. Show two ways to concatenate the following two strings together to get the string "Hi, mom.":**

String hi = "Hi, ";

String mom = "mom.";

**Answer**:

C1: String res = hi.concat(mom)

System.out.println(res);

C2: System.out.println(hi + mom);

**2. Write a program that computes your initials from your full name and displays them.**

import java.util.\*;

public class Ques3 {

    public static void main(String[] args) {

        String name = "   Pham      Duy        Dat        ";

        System.out.println("String: " + name);

        // format String and remove all multiple spaces

        name = name.replaceAll("( )+", " ").trim();

        System.out.println("String: " + name);

        System.out.println("The first Letter Words: " + getinitials(name));

    }

    private static String getinitials(String name) {

        String save[] = name.split(" ");

        String res = "";

        for (String i : save) {

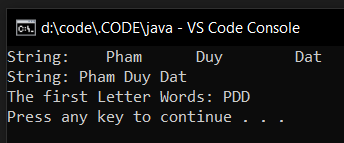
            res += i.charAt(0);

        }

        return res;

    }

}



**3. An anagram is a word or a phrase made by transposing the letters of another word or phrase; for example, "parliament" is an anagram of "partial men," and "software" is an anagram of "swear oft." Write a program that figures out whether one string is an anagram of another string. The program should ignore white space and punctuation.**

import java.util.\*;

public class Ques4 {

    static boolean isAnagram(String A, String B) {

        // A will only keep words and number other special charaters will be removed

        // same as B

        char arrA[] = A.replaceAll("[^a-zA-Z0-9]", "").toLowerCase().toCharArray();

        char arrB[] = B.replaceAll("[^a-zA-Z0-9]", "").toLowerCase().toCharArray();

        Arrays.sort(arrA);

        Arrays.sort(arrB);

        for (int i = 0; i < arrB.length; i++) {

            if (arrA[i] != arrB[i])

                return false;

        }

        return true;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter String 1: ");

        String a = scan.nextLine();

        System.out.println("Enter String 2: ");

        String b = scan.nextLine();

        scan.close();

        System.out.println((isAnagram(a, b)) ? "Is Anagrams" : "Not Anagrams");

    }

}

