1. Write a simple String program to take input from user.
Ans:

```
import java.util.Scanner;

public class Program {
   public static void main(String args[]){
       Scanner scan = new Scanner(System.in);
       String name = scan.nextLine();
       System.out.println("The input name received is " + name + "!");
   }
}
```

2. How do you concatenate two strings in Java? Give an Example?
Ans: Two strings in Java can be concatenated either by using the concat() method or by using the “+” operator.
Example:
For the concat() method:

```
String s = new String("Company");
// From above line of code, a String object s created in the Heap Area //
outside the String constant Pool
System.out.println(s); // This will print Company
s = s.concat(" PW");
// From above line of code, a new String object will be created in the Heap
// Area outside the String constant pool with value Company PW and s will
// now refer to this object
System.out.println(s); // This will print Company PW
```

For the “+” operator:

```
String s1 = "Company";
String s2 = " PW";
String s3 = s1 + s2;
System.out.println(s3); // This will print Company PW
String s4 = "PW" + " Company";
System.out.println(s4); // This will print PW Company
// The difference between s3 and s4 is that for s4, a new object will be
created inside the String Constant Pool(which is inside the Heap Area) and
for s3, eventhough s1 and s2 are created in the String Constant Pool, s3
will be created outside the String Constant Pool because s1 and s2 were
used as reference variables.
```

3. How do you find the length of a string in Java? Explain with an Example?
Ans: The length of a String in Java can be found out using the length() inbuilt method of the String class.
Example:

```
String s1 = "Java Backend DSA Course";
int stringLength = s1.length(); // the method .length() will return the length of the string
System.out.println(stringLength); // prints 23
```

4. How do you compare two strings in Java? Give an Example?
Ans: Two strings in Java can be compared by the following two methods:
i. Using the == operator:
This compares only the references of the two String objects and not the values they hold.
Example:

```
String s1 = "Company"; // s1 will be created in the String Constant Pool
String s2 = "Company"; // Now the JVM scans the String Constant Pool for this object value, i.e, Company and since it is already present it does not create a new object rather s2 wil now point to the same address as s1 since duplicates are not allowed in the String Constant Pool
System.out.println(s1==s2) // This will print true as both s1 and s2 are having the same references
```

```
String s1 = new String("Java"); // s1 will be created in the Heap outside String Constant Pool
String s2 = new String("Java"); // Since duplicates are allowed in the Heap area outside the String Constant Pool, a new object with a new reference will be created in the Heap Area. So, s2 will be having a different reference as compared to s1
System.out.println(s1==s2) // This will print false as s1 and s2 are having different references
```

ii. Using the equals() method
This method compares the values of the two String objects and not their references

```
String s1 = "Company"; // s1 will be created in the String Constant Pool
String s2 =  new String("Company"); // s2 will have a different reference in the Heap Are as it is created using the new keyword
System.out.println(s1.equals(s2)) // This will print true as both s1 and s2 are having the same values eventhough they have different references
```

```
String s1 = new String("Java"); // s1 will be created in the Heap outside String Constant Pool
String s2 = new String("Java"); // Since duplicates are allowed in the Heap area outside the String Constant Pool, a new object with a new reference will be created in the Heap Area. So, s2 will be having a different reference as compared to s1
System.out.println(s1s2) // This will print true as s1 and s2 are having same values eventhough they have different references
```

5. Write a program to find the length of the string "refrigerator".
Ans:

```
import java.util.*;

public class Program {
    public static void main(String args[]){
        String s = "refrigerator";
        int stringLength = s.length();
        System.out.println(stringLength);
    }
}
```

6. Write a program to check if the letter 'e' is present in the word 'Umbrella'.
Ans:

```
import java.util.*;

public class Program {
    public static void main(String args[]){
        String s = "Umbrella";
        boolean isPresent = s.contains("e");
        System.out.println(isPresent);
    }
}
// The above code prints true as e is present in Umbrella
```

7. Write a program to delete all consonants from the String "Hello, have a good day".
Ans:

```
// This program will remove all the consonants including the comma(,) and keep the spacing between the remaining letters as it is
```

```
import java.util.*;

public class Program {
    public static void main(String args[]){
        String originalString = "Hello, have a good day";
        //create a new character array of a small size
        char oCh[] = new char[30];
        int j=0; // for maintaining the index of the char array
        for(int i=0; i< originalString.length(); ++i){
            if(originalString.charAt(i)=='a' ||
originalString.charAt(i)=='e' || originalString.charAt(i)=='i' ||
originalString.charAt(i)=='o' ||
                originalString.charAt(i)=='u' ||
originalString.charAt(i)=='A' || originalString.charAt(i)=='E' ||
originalString.charAt(i)=='I' ||
                originalString.charAt(i)=='O' ||
originalString.charAt(i)=='U'){
                    // If a vowel is found in the String, assign it to the
char array
                    oCh[j] = originalString.charAt(i);
                    // And increase the index of the char array
                    j++;
                }
            else{
                // otherwise add a space
                oCh[j] = ' ';
                j++;
            }
        }
        for(int i=0; i< oCh.length; ++i){
            System.out.print(oCh[i]);
        }
    }
}
```

```
// This program will delete all the consonants and keep no extra space
between the remaining letters
import java.util.*;

public class Program {
```

```
    public static void main(String args[]){
        String originalString = "Hello, have a good day";
        char oCh[] = new char[30];
        int j=0;
        for(int i=0; i< originalString.length(); ++i){
            if(originalString.charAt(i)=='a' ||
originalString.charAt(i)=='e' || originalString.charAt(i)=='i' ||
originalString.charAt(i)=='o' ||
                originalString.charAt(i)=='u' ||
originalString.charAt(i)=='A' || originalString.charAt(i)=='E' ||
originalString.charAt(i)=='I' ||
                originalString.charAt(i)=='O' ||
originalString.charAt(i)=='U'){
                    oCh[j] = originalString.charAt(i);
                    j++;
                }
        }
        for(int i=0; i< oCh.length; ++i){
            System.out.print(oCh[i]);
        }
        System.out.println();
    }
}
```