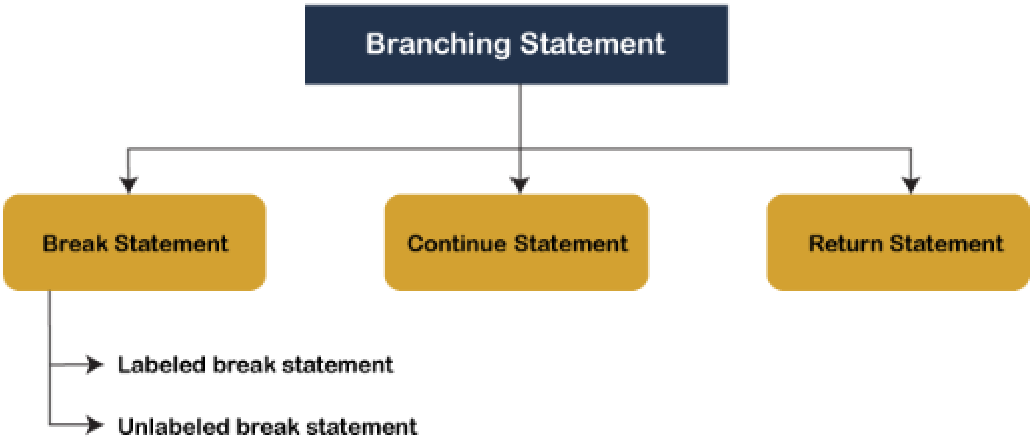
Name: Shaikh Salif Aminuddin

CLASS: S2 **ROLL NO:** 2201094

EXPERIMENT NO: 1

AIM: To Implement and Learn Programs on Basic programming constructs like branching and looping.

THEORY: Branching statements are the statements used to jump the flow of execution from one part of a program to another. The branching statements are mostly used inside the control statements. Java has mainly three branching statements, i.e., continue, break, and return. The branching statements allow us to exit from a control statement when a certain condition meet.



The break statement breaks or terminates the loop and transfers the control outside the loop. The continue statement skips the current execution and pass the control to the start of the loop. The return statement returns a value from a method and this process will be done explicitly.

The continue statement is another branching statement used to immediately jump to the next iteration of the loop. It is a special type of loop which breaks current iteration when the condition is met and start the loop with the next iteration. In simple words, it continues the current flow of the program and stop executing the remaining code at the specified condition.

The return statement is also a branching statement, which allows us to explicitly return value from a method. The return statement exits us from the calling method and passes the control flow to where the calling method is invoked. Just like the break statement, the return statement also has two forms, i.e., one that passes some value with control flow and one that doesn't.

Loops in Java

The Java for loop is used to iterate a part of the program several times. If the number of iteration is fixed, it is recommended to use for loop. There are three types of for loops in Java.

1. Simple for Loop
2. For-each or Enhanced for Loop
3. Labeled for Loop

Java Simple for Loop

A simple for loop is the same as C/C++. We can initialize the variable, check condition and increment/decrement value. It consists of four parts:

Inititialization - Condition - Increment/decrement - statement

Java Nested for Loop

If we have a for loop inside the another loop, it is known as nested for loop. The inner loop executes completely whenever outer loop executes.

Java for-each Loop

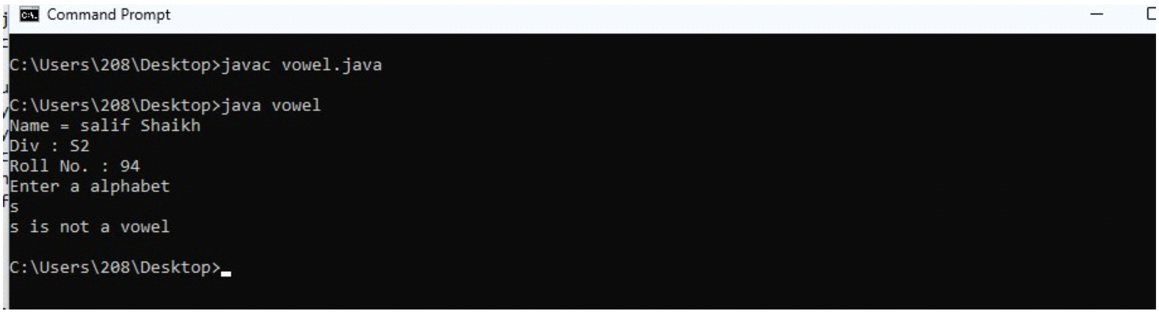
The for-each loop is used to traverse array or collection in Java. It is easier to use than simple for loop because we don't need to increment value and use subscript notation. It works on the basis of elements and not the index. It returns element one by one in the defined variable.

CODE:

```
import java.util.Scanner;

public class vowel {

        public static void main(String[] args){

        System.out.println("Name = Shaikh Salif\nDiv : S2\nRoll No. : 94");
```

```
    System.out.println("Enter a alphabet");
    Scanner sc = new Scanner(System.in);
    char data = sc.next().charAt(0);
    if(data == 'A'||data=='E'||data=='I'||data=='O'||data=='U'||
      data == 'a'||data=='e'||data=='i'||data=='o'||data=='u')
    {
        System.out.println(data + " is a vowel");
    }
    else
    {
        System.out.println(data + " is not a vowel");
    }
    }
}
```

OUTPUT:



```
C:\Users\208\Desktop>javac vowel.java

C:\Users\208\Desktop>java vowel
Name = salif Shaikh
Div : S2
Roll No. : 94
Enter a alphabet
s
s is not a vowel

C:\Users\208\Desktop>_
```

CONCLUSION: We have now completely understood and applied basics of java programming like branching and loops**.**