Batch: B2	Roll No.: 16010124107
Experiment / assignment / tutorial No2	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE : Control Statements

AIM:

Write a Java program to generate and show all Kaprekar numbers less than 1000.
In number theory, a Kaprekar number for a given base is a non-negative integer, the representation of whose square in that base can be split into two parts that add up to the original number again. For instance, 45 is a Kaprekar number, because 452 = 2025 and 20 + 25 = 45.

Expected OUTCOME of Experiment:

CO1:Apply the features of object oriented programming languages. (C++ and Java)
CO2:Explore arrays, vectors, classes and objects in C++ and Java

Books/ Journals/ Websites referred:

1. E. Balagurusamy, “Programming with Java”, McGraw-Hill.
2. E. Balagurusamy, “Object Oriented Programming with C++”, McGraw-Hill.

Pre Lab/ Prior Concepts:

Java basic constructs (like if else statement, control structures, and data types
Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of

statements multiple times and following is the general form of a loop statement in most of the programming languages −

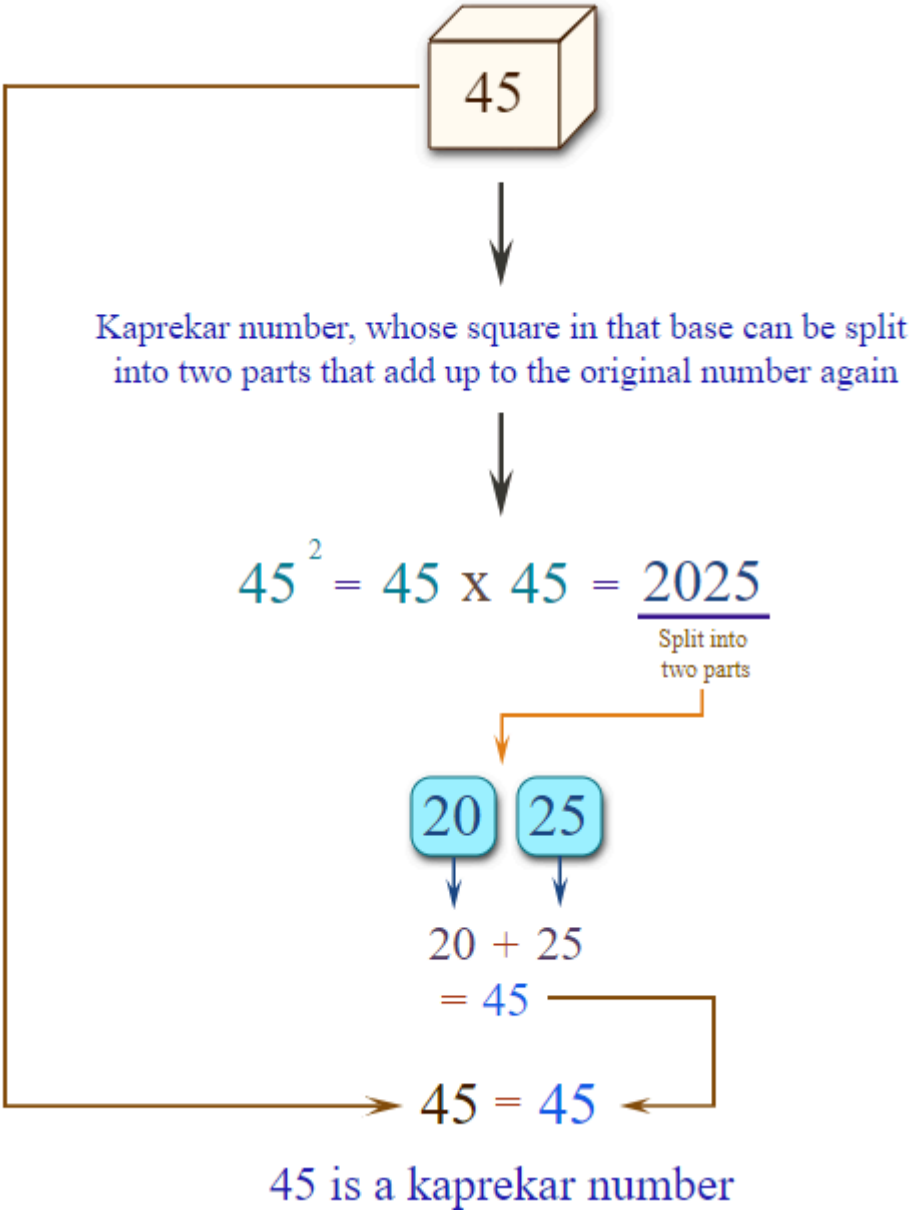
Sr.No.	Loop & Description
1	**while loop** Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
2	**for loop** Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.
3	**do...while loop** Like a while statement, except that it tests the condition at the end of the loop body.

Loop Control Statements

Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.

Java supports the following control statements. Click the following links to check their details.

Sr.No.	Control Statement & Description
1	**break statement** Terminates the loop or switch statement and transfers execution to the statement immediately following the loop or switch.
2	**continue statement** Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.



In number theory, a Kaprekar number for a given base is a non-negative integer, the representation of whose square in that base can be split into two parts that add up to the original number again. For instance, 45 is a Kaprekar number, because 45^2 = 2025 and 20 + 25 = 45.

Algorithm:

1. Start
2. Import scanner
3. Initialise class and main

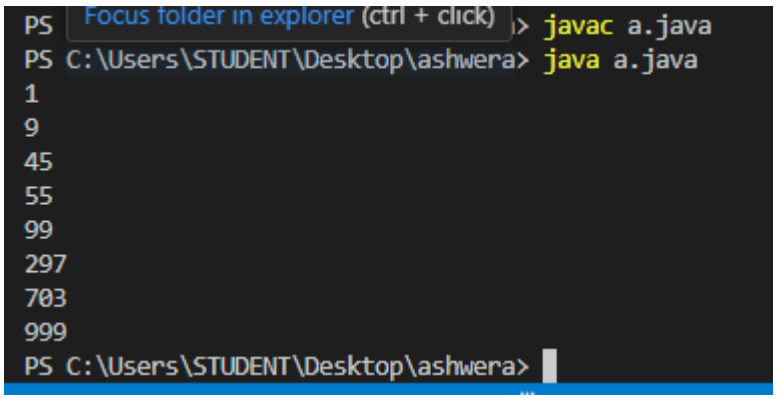
4. Run loop from i=0 to 1000
5. Find i's square
6. Extract lhs and rhs of square
7. Add and check if its equal to original number
8. If yes, print
9. Continue for 1000 iterations
10. Stop

Implementation details:

```
import java.util.*;


class a {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        for(int i=1;i<1000;i++)

        {

            int sq = i*i;

        int d = String.valueOf(i).length();

        int divisor = (int) Math.pow(10, d);


        int r = sq % divisor;

        int l = sq / divisor;


        if (r + l == i && r != 0) {

            System.out.println(i);

        }
```

```
        }

    }

}
```

Output:



Conclusion:

In Java, the break and continue statements help us control the flow of the loops. We can use break to terminate our block early and we can use continue to skip over the statements under certain conditions. Using these, we can optimise codes and deal with conditions where we do not know the condition beforehand.

Date: 28/07/2025 **Signature of faculty in-charge**

Post Lab Descriptive Questions:

Q.1 Write a program to find the largest of three numbers using the if-else construct.
code:

import java.util.*;

```
class a
{
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter 3 numbers: ");
      int n = sc.nextInt();
      int n1 = sc.nextInt();
      int n2 = sc.nextInt();
      if(n>n1)
      {
         if(n>n2)
         {
            System.out.println(n+" is the largest number");
         }
         else if(n2>n1)
         {
            System.out.println(n2+" is the largest number");
         }
      }
      else if(n1>n2)
      {
         System.out.println(n1+" is the largest number");
      }
   }
}
```

Q.2 Write a program to determine the sum of the following series for a given value of n:1+½+⅓+....+1/n

```
 import java.util.*;
class a
{
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter number: ");
      int n = sc.nextInt();
      float sum=0;
      for (float i = 1; i <= n ; i++){
```

```
            sum += ((float)1/i);
        }
        System.out.println("Sum of series is: "+sum);
    }
}
```

Output:

1.
```
PS C:\Users\STUDENT\Desktop\ashwera> javac a.java
PS C:\Users\STUDENT\Desktop\ashwera> java a.java
Enter 3 numberS:
5 1 95
95 is the largest number
PS C:\Users\STUDENT\Desktop\ashwera>
```

2.
```
PS C:\Users\STUDENT\Desktop\ashwera> javac a.java
PS C:\Users\STUDENT\Desktop\ashwera> java a.java
Enter number:
5
Sum of series is: 2.2833335
```