Ex.No:1(A) CLASS & OBJECTS

AIM:

To create a class named 'Student' with String variable 'name' and String variable 'address'.

ALGORITHM :

1. Start the program.
2. Define a class named 'Student'
3. Declare a String variable 'name' and initialize it with the value "John"
4. Declare a String variable 'address' and initialize it with the value "Chennai"
5. Define a class named 'Test'
6. Define the 'main' method within the 'Test' class
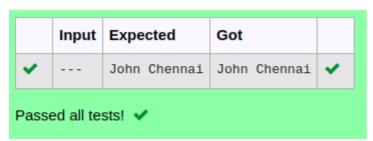7. Create an object 'obj' of the 'Student' class
8. Print the value of 'name' and 'address' variables of the 'obj' object
9. End

PROGRAM:

```
Program to implement a class & objects using Java
Developed by    : Sam Israel D
RegisterNumber  : 212222230128
```

Sourcecode.java:

```java
class Student {
    String name = "John";
    String address = "Chennai";
}

class Test {
    public static void main(String[] args) {
        Student obj = new Student();
        System.out.println(obj.name);
        System.out.println(obj.address);
    }
}
```

OUTPUT:

	Input	Expected	Got	
✔	---	John Chennai	John Chennai	✔

Passed all tests! ✔

RESULT:

Thus, the class named 'Student' with String variable 'name' and String variable 'address' was created successfully.

Ex.No:1(B) VARIABLES AND OPERATOR

AIM:

To write a Java program to get values of variables 'a' and 'b' and then check if both the conditions 'a < 50' and 'a < b' are true. [Class name is 'Demo']

ALGORITHM :

1. Start the program.

2. Import the necessary package 'java.util'

3. Define a class named 'Demo'

4. Implement the main method

5. Create a new instance of the 'Scanner' class named 'sc' to read user input

6. Read an integer 'a' from the user using the 'nextInt' method of 'sc'

7. Read another integer 'b' from the user using the 'nextInt' method of 'sc'

8. Check if 'a' is less than 50 or if 'a' is less than 'b'
   - If the condition is true, print "true" using the 'print' method of 'System.out'
   - If the condition is false, print "false" using the 'print' method of 'System.out'

9. End

PROGRAM:

```
Program to implement a variable and operators using Java
Developed by     : Sam Israel D
RegisterNumber   : 212222230128
```

Sourcecode.java:

```java
import java.util.*;
public class Demo
{
    public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    double a = sc.nextDouble();
    double b = sc.nextDouble();
    System.out.println(a < 50 || a < b);
    }
}
```

OUTPUT:

	Input	Expected	Got	
✔	23 50	true	true	✔
✔	50 65	true	true	✔
✔	55 45	false	false	✔

Passed all tests! ✔

RESULT:

Thus, the Java program to get values of variables 'a' and 'b' and then check if both the conditions 'a < 50' and 'a < b' are true is created successfully.

Ex.No:1(C) CONTROL STATEMENTS

AIM:

To develop a Java program to check given number is zero or not.

ALGORITHM :

1. Start the program.

2. Declare an integer variable 'num'

3. Create a Scanner object 'sc' to read input from the user

4. Read an integer input from the user and store it in 'num'

5. Check if 'num' is equal to 0: a. If true, print "Given number is Zero" b. If false, print 'num' followed by " is Non-Zero"

6. End

PROGRAM:

```
Program to implement a class & objects using Java
Developed by     : Sam Israel D
RegisterNumber   : 212222230128
```

Sourcecode.java:

```java
import java.util.Scanner;

public class Demo
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int num=sc.nextInt();
         if(num == 0)
         System.out.println("Given number is Zero");
         else
         System.out.format("%d is Non-Zero",num);
    }
}
```

OUTPUT:

	Input	Expected	Got	
✔	0	Given number is Zero	Given number is Zero	✔
✔	12	12 is Non-Zero	12 is Non-Zero	✔
✔	-32	-32 is Non-Zero	-32 is Non-Zero	✔

Passed all tests! ✔

RESULT:

Thus, the Java program to check given number is zero or not was created successfully.

Ex.No:1(D) USER DEFINED METHOD.

AIM:

To create a Java program print area of rectangle by defining instance method and local variable value as 10,20 .[Class Name is 'Area' function name is 'calculateArea()' and return type of function is 'void'

ALGORITHM :

1. Start the program.
2. Define a class named 'Area'
3. Declare a public method named 'calculateArea' with no parameters
4. Inside the 'calculateArea' method:
   - Declare a Double variable 'length' and assign it the value 10.0
   - Declare a Double variable 'width' and assign it the value 20.0
   - Calculate the area by multiplying 'length' and 'width' and store the result in a Double variable 'area'
   - Print the calculated area using the System.out.println statement
5. Define the 'main' method as static
6. Inside the 'main' method:
   - Create an instance of the 'Area' class called 'rectangle'
   - Call the 'calculateArea' method on the 'rectangle' object

PROGRAM:

```
Program to implement a User Defined Method using Java
Developed by    : Sam Israel D
RegisterNumber  : 212222230128
```

Sourcecode.java:

```java
import java.util.*;
public class Area {
        void calculateArea()
    {
        double side,cirarea;
        Scanner sc=new Scanner(System.in);
        side=sc.nextDouble();
        cirarea=side*side;
        System.out.println("Area of Square is "+cirarea);
    }
    public static void main(String[] args) {
        Area obj = new Area();
        obj.calculateArea();
        }
}
```

OUTPUT:

	Input	Expected	Got	
✔	2	Area of Square is 4.0	Area of Square is 4.0	✔
✔	2.5	Area of Square is 6.25	Area of Square is 6.25	✔
✔	1.0	Area of Square is 1.0	Area of Square is 1.0	✔

Passed all tests! ✔

RESULT:

Thus, the Java program to print area of rectangle by defining instance method and local variable value as 10,20 was created successfully.

Ex.No:1(E) INSTANCE METHOD AND VARIABLE

AIM:

To write a Java program to check whether the number 2003 is prime or not using an instance method and instance variable. The class is named Prime, and the method isPrime() returns a boolean indicating whether the number is prime.

ALGORITHM :

1. Start the program.
2. Create a class named `Prime`.
3. Declare an instance variable `number` in the class.
4. Define a constructor to initialize the `number` variable.
5. Create an instance method `isPrime()` with return type `boolean`.
6. In the `isPrime()` method:
   - Check if the number is less than or equal to 1. If so, return false.
   - Use a loop to check for factors from 2 up to the square root of the number.
   - If any factor divides the number evenly, return false.
   - If no factors are found, return true.
7. In the `main` method:
   - Create an object of the `Prime` class, initializing it with the number 2003.
   - Call the `isPrime()` method using the object.
   - Print whether the number is prime based on the method's return value.
8. End the program.

PROGRAM:

```
Program to implement a Static Variable using Java
Developed by     : Sam Israel D
RegisterNumber   : 212222230128
```

Sourcecode.java:

```java
import java.util.*;
public class Prime {
    int num = 2003, count=0;
    boolean isPrime() {
        for(int i=1; i<=num/2;i++) {
            if (num%i==0) {
                count+=1;
            }
        }
        return (count==1);
    }
    public static void main(String[] args) {
        Prime obj = new Prime();
        System.out.println(obj.isPrime());
    }
}
```

OUTPUT:

	Expected	Got	
✔	true	true	✔

Passed all tests! ✔

RESULT:

Thus, the Java program for the concept of using a static variable for shared data was correctly implemented and verified successfully.