Pradnya Hitendra More

PRN: 2502405200587

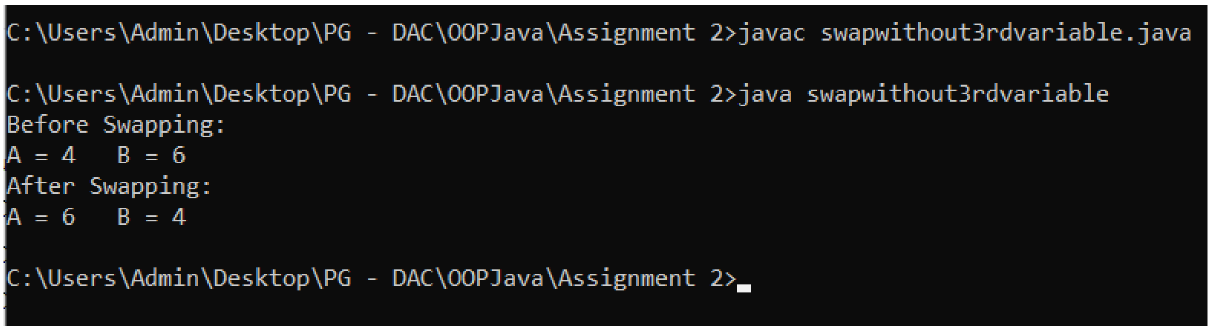
Form No.: 241205579

1. Arithmetic & Assignment Operators

Q1: Write a program to swap two numbers without using a third variable and without using arithmetic operators like + or - .

Hint : Use bitwise XOR ^ operator.

```
public class swapwithout3rdvariable{
        public static void main(String[] args){
                int a = 4;
                int b = 6;
                System.out.println("Before Swapping: \nA = " + a + "\tB = " + b);
                a = a ^ b;
                b = a ^ b;
                a = a ^ b;
                System.out.println("After Swapping: \nA = " + a + "\tB = " + b);
        }
}
```

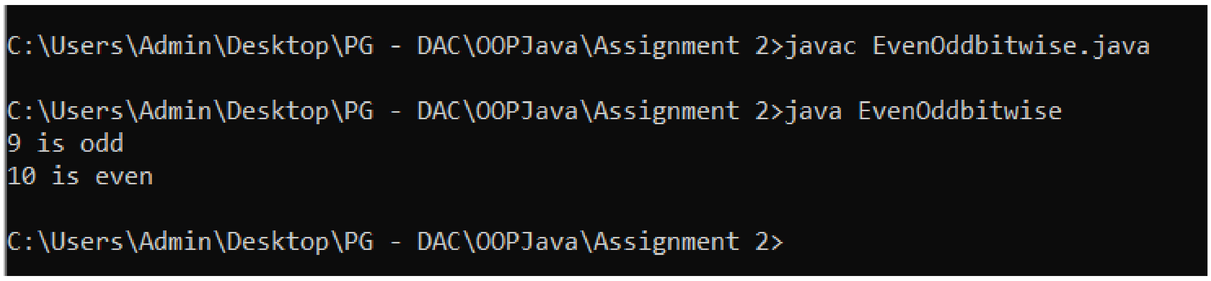


```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac swapwithout3rdvariable.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java swapwithout3rdvariable
Before Swapping:
A = 4   B = 6
After Swapping:
A = 6   B = 4

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>
```

Q2: Write a program to check whether a given number is even or odd using only bitwise operators .

Hint : Use n & 1 to check.

```
public class EvenOddbitwise{
        public static void main(String[] args){
                int a = 9, b = 10;
                check(a);
                check(b);
        }
         public static void check(int a){
                if((a&1) == 0){
                        System.out.println(a + " is even");
                } else {
                        System.out.println(a + " is odd");
                }
        }
}
```

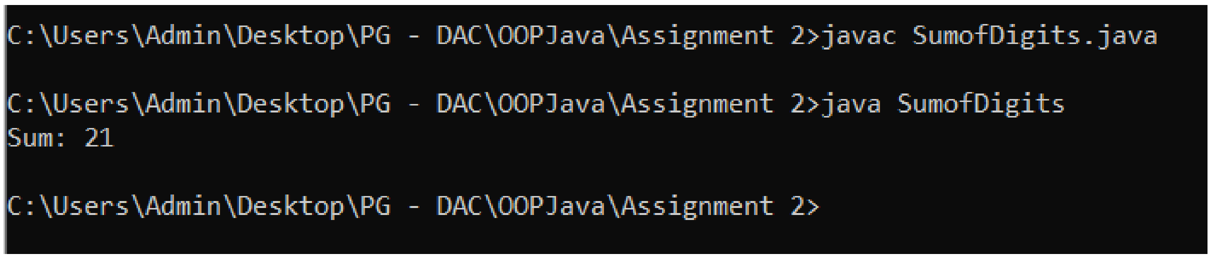


```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac EvenOddbitwise.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java EvenOddbitwise
9 is odd
10 is even

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>
```

Q3: Implement a program that calculates the sum of digits of an integer using modulus (%) and division (/) operators .

```
public class SumofDigits{
        public static void main(String[] args){
```

```
        int x = 123456;
        int res = 0;
        while(x!=0){
            res = res + (x%10);
            x = x/10;
        }
        System.out.println("Sum: " + res);
    }
}
```



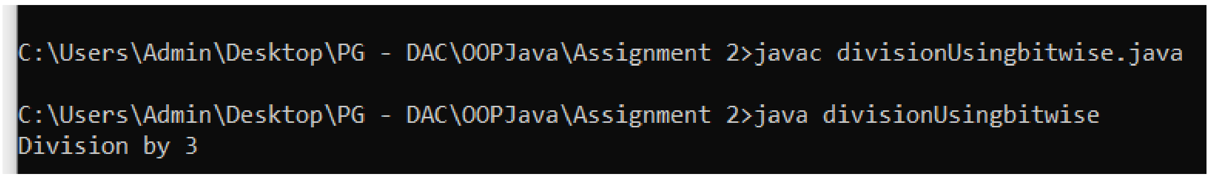
```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac SumofDigits.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java SumofDigits
Sum: 21

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>
```

Q4: Write a program to find whether a given number is divisible by 3 without using the modulus (%) or division (/) operators.

Hint : Use subtraction and bitwise shifts .

```
public class divisionUsingbitwise{
    public static void main(String[] args){
        int n = 15;

 // Convert the number to positive if it's negative
        if(n<0){
            n = -n;
        }

   // Use bitwise operations and subtraction to check divisibility by 3
        while(n>3){
            n = (n & 3) + (n >> 2);// Reduce n by using bitwise operations
```

```
                    }

                    // Check if the result is 0 or 3, meaning the number is divisible by 3
                    if(n==0 || n==3)
                            System.out.println("Division by 3");
                    else
                            System.out.println("Not division by 3");
            }
}
```

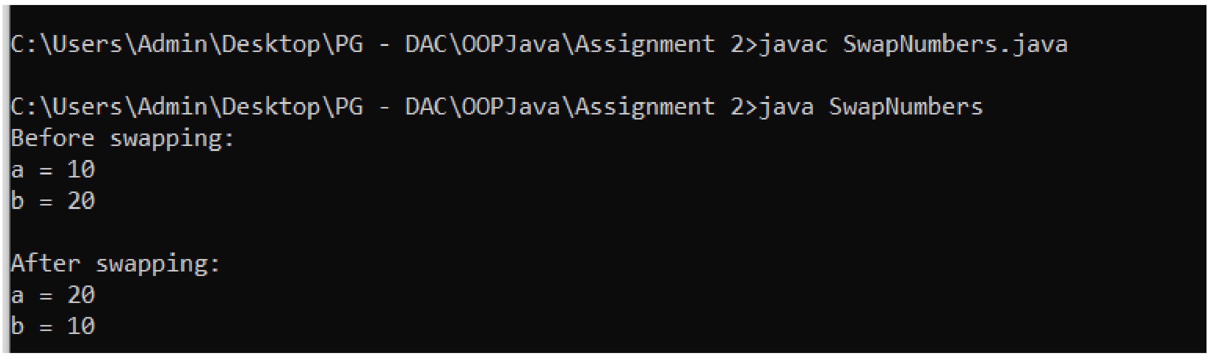


```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac divisionUsingbitwise.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java divisionUsingbitwise
Division by 3
```

Q5: Write a Java program to swap two numbers using the += and -= operators only.

```
public class SwapNumbers {
   public static void main(String[] args) {
      // Initializing two numbers
      int a = 10;
      int b = 20;
      System.out.println("Before swapping:");      // Displaying the numbers before swapping
      System.out.println("a = " + a);
      System.out.println("b = " + b);

      // Swapping using += and -= operators
      a = a + b;  // a now becomes the sum of a and b
      b = a - b;  // b is now the original value of a
      a = a - b;  // a is now the original value of b
```

```
        // Displaying the numbers after swapping
        System.out.println("\nAfter swapping:");
        System.out.println("a = " + a);
        System.out.println("b = " + b);
    }
}
```

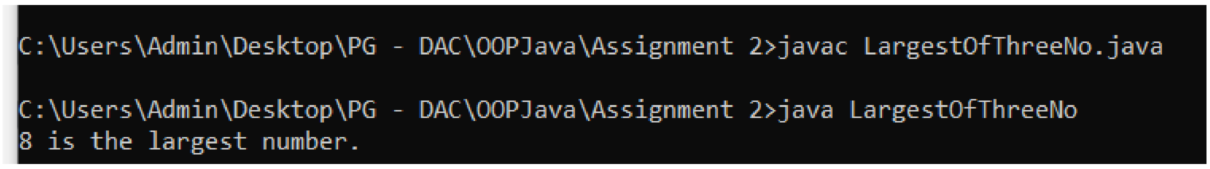


```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac SwapNumbers.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java SwapNumbers
Before swapping:
a = 10
b = 20

After swapping:
a = 20
b = 10
```

2. Relational & Logical Operators

Q6: Write a program to find the largest of three numbers using only the ternary operator (? :) .

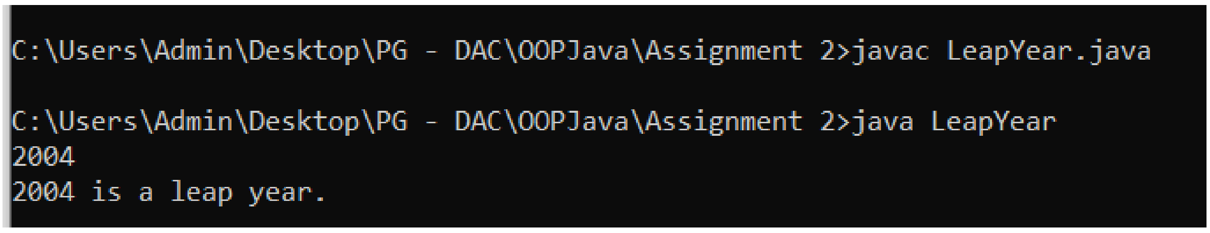
```
public class LargestOfThreeNo {
    public static void main(String[] args) {
        // Initializing the three variables with values
        int x = 8, y = 4, z = 1;

        // Using a ternary operator to find the largest number among x, y, and z
        // First, check if x is greater than both y and z
        int res = ((x > y && x > z) ? x : // If x is largest, assign x to res
                  (y > x && y > z) ? y : // If y is largest, assign y to res
                  z);  // If neither x nor y is largest, assign z to res
        // Print the largest number
        System.out.println(res + " is the largest number.");
    }
}
```



```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac LargestOfThreeNo.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java LargestOfThreeNo
8 is the largest number.
```

Q7: Implement a Java program that checks whether a given year is a leap year or not using logical (&& , ||) operators .

```
import java.util.Scanner;  // Importing the Scanner class to read input from the user
public class LeapYear {
    public static void main(String[] args) {
        // Create a Scanner object to take input from the user
        Scanner sc = new Scanner(System.in);

        // Read an integer input from the user, which represents the year
        int y = sc.nextInt();

        // Check if the year is a leap year using the given conditions
        // A year is a leap year if:
        // - It is divisible by 400, or
        // - It is divisible by 4 but not divisible by 100
        if (y % 400 == 0 || (y % 4 == 0 && y % 100 != 0))
            // If the conditions are true, print that the year is a leap year
            System.out.println(y + " is a leap year.");
        else
            // If the conditions are false, print that the year is not a leap year
            System.out.println(y + " is not a leap year.");
    }
}
```

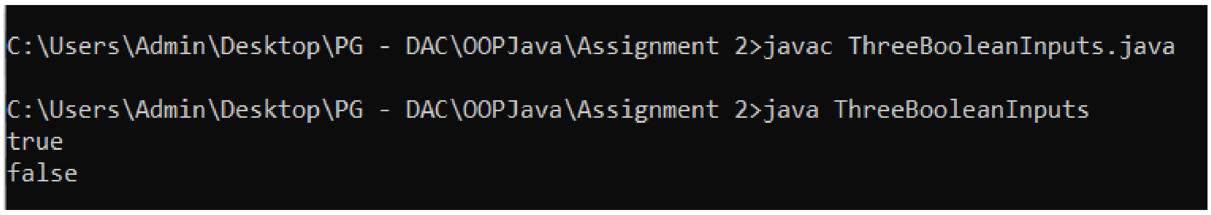


```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac LeapYear.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java LeapYear
2004
2004 is a leap year.
```

Q8: Write a program that takes three boolean inputs and prints true if at least two of them are true .

Hint : Use logical operators (&& , ||).

```
public class ThreeBooleanInputs {
    public static void main(String[] args) {
        // Initializing three boolean variables x, y, and z with initial values
        boolean x = true, y = false, z = true;

        // Calling the check method with x, y, and z as parameters
        check(x, y, z);  // First call: x = true, y = false, z = true

        // Changing the values of x, y, and z
        x = true; y = false; z = false;

        // Calling the check method again with the updated values of x, y, and z
        check(x, y, z);  // Second call: x = true, y = false, z = false
    }
    // Method that checks if at least two boolean values are true
    public static void check(boolean a, boolean b, boolean c) {
        // The condition checks if at least two of the three boolean values are true
        // Using logical operators (&& for AND, || for OR)
        if ((a && (b || c)) || (c && (a || b)) || (b && (a || c))) {
            // If the condition evaluates to true, print "true"
            System.out.println("true");
        } else {
            // If the condition evaluates to false, print "false"
            System.out.println("false");
        }
    }
}
```

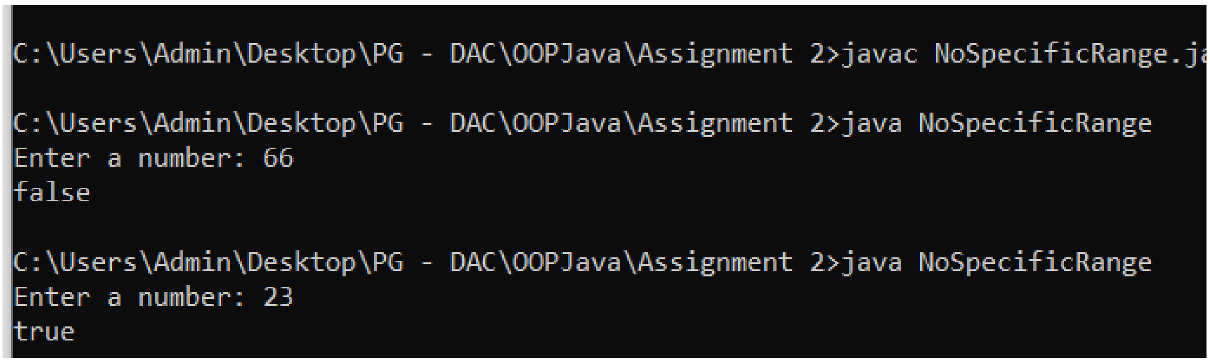


```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac ThreeBooleanInputs.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java ThreeBooleanInputs
true
false
```

Q9: Implement a Java program that checks if a number is within a specific range (20 to 50) without using if-else .

Hint : Use logical AND (&&) in a print statement .

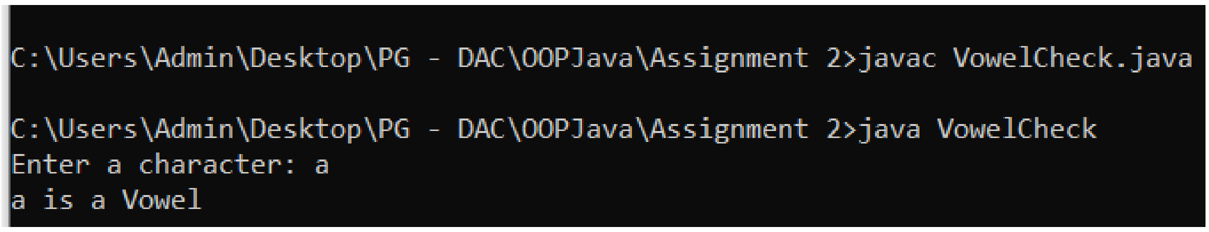
```
import java.util.Scanner;

public class NoSpecificRange {
    public static void main(String[] args) {
        // Create a Scanner object to take input from the user
        Scanner sc = new Scanner(System.in);


        // Prompt the user to enter a number

        System.out.print("Enter a number: ");

        int num = sc.nextInt();


        // Print the result based on the condition

        System.out.println(num >= 20 && num <= 50 && "Number is within the range (20 to
50).".equals("Number is within the range (20 to 50).") );


    }

}
```



```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac NoSpecificRange.ja

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java NoSpecificRange
Enter a number: 66
false

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java NoSpecificRange
Enter a number: 23
true
```

Q10: Write a program to determine if a character is a vowel or a consonant using the ternary operator.

```
import java.util.Scanner;

public class VowelCheck {

    public static void main(String[] args) {

        // Create a Scanner object to take input from the user

        Scanner sc = new Scanner(System.in);


        // Prompt the user to enter a character

        System.out.print("Enter a character: ");
        char ch = sc.next().charAt(0);


        // Use the ternary operator to check if the character is a vowel or consonant
        String result = (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
                         ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
                         ? "Vowel" : "Consonant";


        // Print the result
        System.out.println(ch + " is a " + result);
         }
}
```



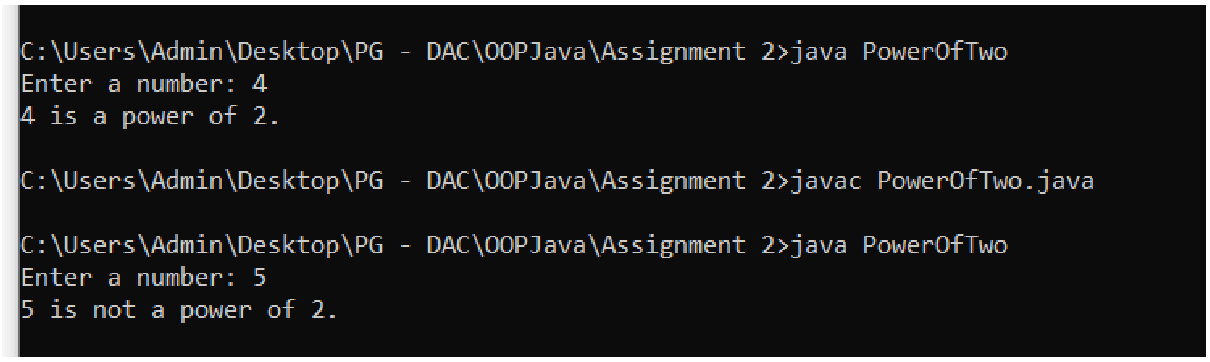
```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac VowelCheck.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java VowelCheck
Enter a character: a
a is a Vowel
```

3. Bitwise Operators

Q11: Write a program to check if a given number is a power of 2 using bitwise operators.

Hint : n & (n - 1) == 0 for positive numbers.

```
import java.util.Scanner;


public class PowerOfTwo {
    public static void main(String[] args) {
        // Create a Scanner object to take input from the user
        Scanner sc = new Scanner(System.in);


        // Prompt the user to enter a number
        System.out.print("Enter a number: ");
        int n = sc.nextInt();

        // Check if the number is a power of 2 using the bitwise operator
        boolean isPowerOfTwo = (n > 0) && (n & (n - 1)) == 0;

        // Output the result
        if (isPowerOfTwo) {
            System.out.println(n + " is a power of 2.");
        } else {
            System.out.println(n + " is not a power of 2.");
        }

        // Close the scanner
        sc.close();
    }
}
```



```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java PowerOfTwo
Enter a number: 4
4 is a power of 2.

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac PowerOfTwo.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java PowerOfTwo
Enter a number: 5
5 is not a power of 2.
```

Q12: Write a Java program to multiply a number by 8 without using * or / operators.

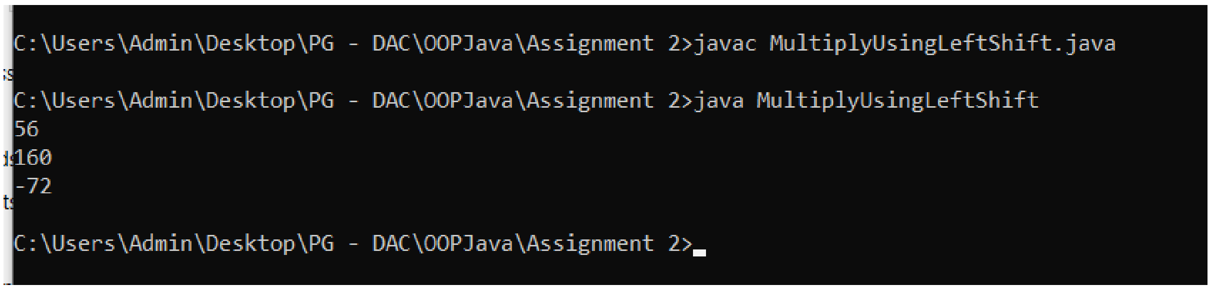
Hint : Use bitwise left shift (<<).

```
public class Q12 {
    public static void main(String[] args) {
        // Calling the check method with different integer values
        check(7);   // Will multiply 7 by 8 using left shift
        check(20);  // Will multiply 20 by 8 using left shift
        check(-9);  // Will multiply -9 by 8 using left shift
    }

    // Method to multiply a number by 8 using bitwise left shift (<<)
    public static void check(int a) {
        // Perform a left shift by 3 positions (equivalent to multiplying by 8)
        int result = a << 3;

        // Print the result of the left shift operation
        System.out.println(result);
    }
}
```

Q13: Implement a Java program to find the absolute value of an integer using bitwise operators.

Hint : mask = num >> 31; abs = (num + mask) ^ mask;

```
public class MultiplyUsingLeftShift {
    public static void main(String[] args) {
        // Calling the check method with different integer values
        check(7);   // Will multiply 7 by 8 using left shift
        check(20);  // Will multiply 20 by 8 using left shift
        check(-9);  // Will multiply -9 by 8 using left shift
    }
    // Method to multiply a number by 8 using bitwise left shift (<<)
    public static void check(int a) {
        // Perform a left shift by 3 positions (equivalent to multiplying by 8)
        int result = a << 3;

        // Print the result of the left shift operation
        System.out.println(result);
    }
}
```

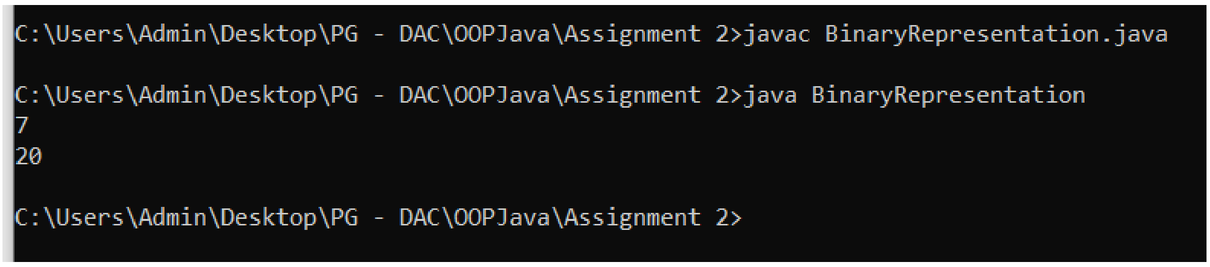


```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac MultiplyUsingLeftShift.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java MultiplyUsingLeftShift
56
160
-72

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>
```

Q14: Write a program to count the number of 1s (set bits) in a binary representation of a number using bitwise operations.

Hint : Use n & (n - 1) .

```
public class BinaryRepresentation  {
    public static void main(String[] args) {
        // Calling the abs method with both positive and negative integers
        abs(7);   // Test with a positive number
        abs(-20); // Test with a negative number
    }

    // Method to compute the absolute value of a number using bitwise operations
    public static void abs(int num) {
        // Create a mask by right-shifting the number by 31 bits (sign bit)
        // If num is positive, mask = 0 (no effect); if num is negative, mask = -1 (0xFFFFFFFF)
        int mask = num >> 31;

        // Compute the absolute value by adjusting num if it is negative
        // If num is positive, the mask is 0, and (num + mask) ^ mask is just num.
        // If num is negative, the mask is -1, and (num + mask) becomes (num - 1), then ^ mask inverts the result.
        int abs = (num + mask) ^ mask;

        // Print the computed absolute value
        System.out.println(abs);
    }
}
```

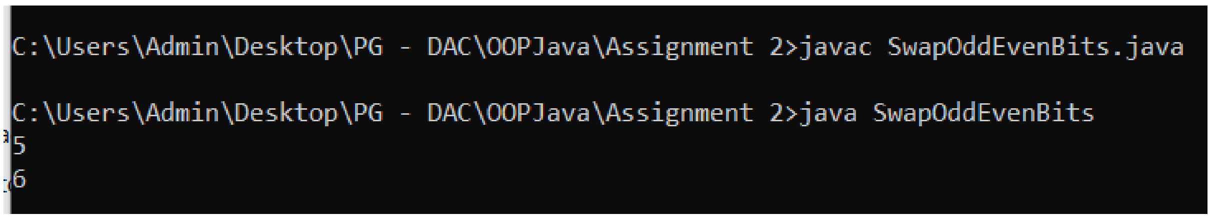


```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac BinaryRepresentation.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java BinaryRepresentation
7
20

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>
```

Q15: Implement a program to swap odd and even bits of a number using bitwise operators.

Hint : Use masks: (x & 0xAAAAAAAA) >> 1 | (x & 0x55555555) << 1

```
public class SwapOddEvenBits {
    public static void main(String[] args) {
        // Test cases: Calling swapper method with different numbers
        swapper(10); // 10 in binary: 00001010
        swapper(9);  // 9 in binary: 00001001
    }

    // Method to swap odd and even bits of the number
    public static void swapper(int x) {
        // Get even bits (bit positions 0, 2, 4, 6...) and shift them right by 1 position
        int evenBits = (x & 0xAAAAAAAA) >> 1;
        // Get odd bits (bit positions 1, 3, 5, 7...) and shift them left by 1 position
        int oddBits = (x & 0x55555555) << 1;

        // Combine the shifted even and odd bits using bitwise OR
        int res = (evenBits | oddBits);

        // Print the result of the bit swap
        System.out.println(res);
    }
}
```

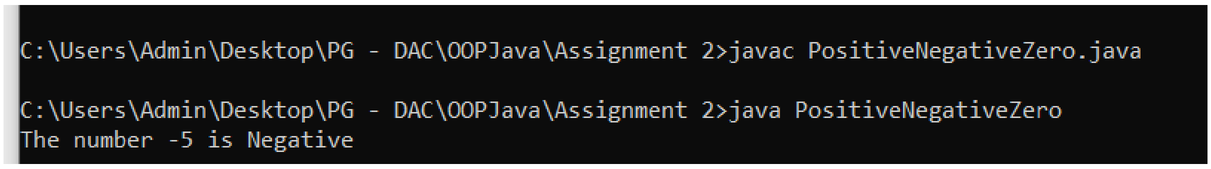


```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac SwapOddEvenBits.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java SwapOddEvenBits
5
6
```

4. Ternary Operator Challenges

Q16: Write a program that determines whether a given number is positive, negative, or zero using only the ternary operator .

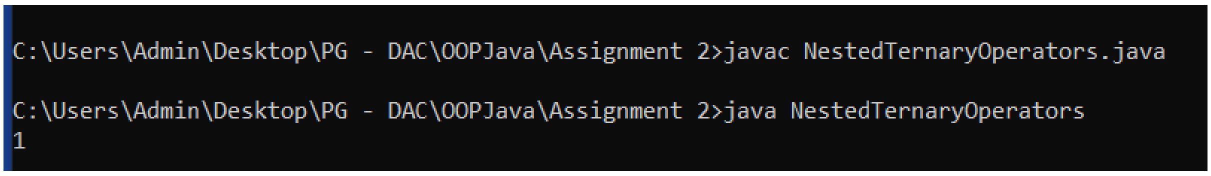
```
public class PositiveNegativeZero {
    public static void main(String[] args) {
        int num = -5; // You can change this value to test with different numbers

        // Using ternary operator to determine if the number is positive, negative, or zero
        String result = (num > 0) ? "Positive" : (num < 0) ? "Negative" : "Zero";

        // Print the result
        System.out.println("The number " + num + " is " + result);
    }
}
```



```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac PositiveNegativeZero.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java PositiveNegativeZero
The number -5 is Negative
```

Q17: Implement a Java program that finds the minimum of four numbers using nested ternary operators.

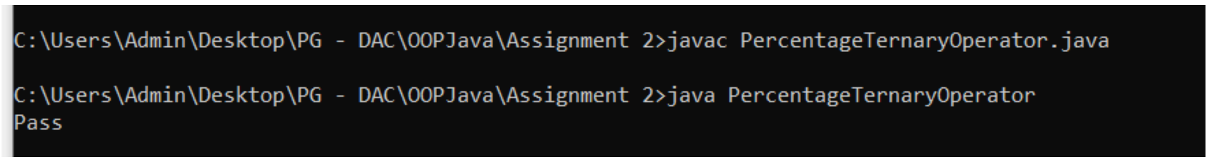
```
public class NestedTernaryOperators {
    public static void main(String[] args) {
        // Calling the minfour method with four integer arguments
        minfour(1, 2, 3, 4);
    }

    // Method to find the minimum of four numbers using nested ternary operators
    public static void minfour(int x, int y, int z, int m) {
        // Using nested ternary operators to find the minimum of the four numbers
        int res = (x < y && x < z && x < m) ? x : // Check if x is the smallest
                  (z < y && z < x && z < m) ? z : // If x is not the smallest, check if z is the smallest
                  (y < z && y < x && y < m) ? y : // If neither x nor z is the smallest, check if y is the
smallest
                  m;  // If none of the above are the smallest, m must be the smallest

        // Print the minimum value
        System.out.println(res);
    }
}
```



```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac NestedTernaryOperators.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java NestedTernaryOperators
1
```

Q18: Given a student's percentage, print "Pass" if the percentage is 40 or above; otherwise, print "Fail" , using only the ternary operator.

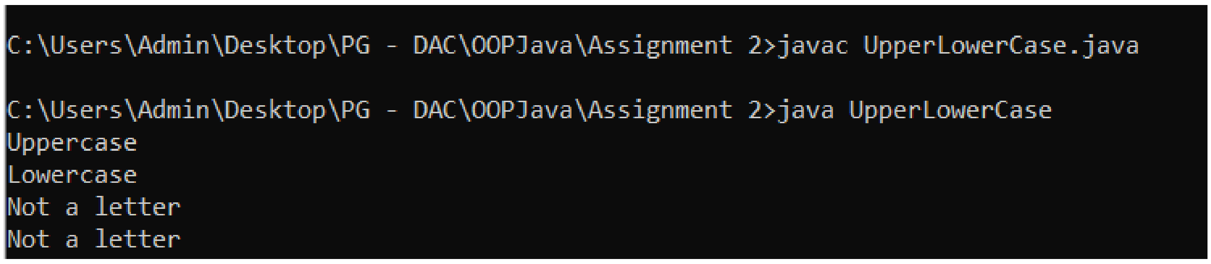
```
public class  PercentageTernaryOperator {
    public static void main(String[] args) {
        double percentage = 45.5; // You can change this value to test with different
percentages

        // Using the ternary operator to check if the percentage is 40 or above
        String result = (percentage >= 40) ? "Pass" : "Fail";

        // Print the result
        System.out.println(result);
    } }
```



```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac PercentageTernaryOperator.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java PercentageTernaryOperator
Pass
```

Q19: Write a Java program that checks whether a character is uppercase, lowercase, or not a letter using only the ternary operator.

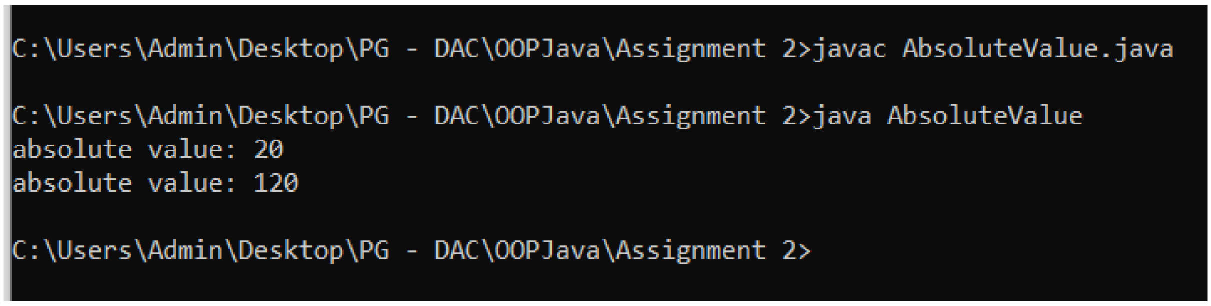
```
public class UpperLowerCase {
    public static void main(String[] args) {
        // Test characters
        char ch1 = 'A';  // Uppercase character
        char ch2 = 'z';  // Lowercase character
        char ch3 = '1';  // Non-letter character
        char ch4 = '!';  // Non-letter character

        // Check and print the result using ternary operators
        System.out.println(checkCharacterCase(ch1)); // Uppercase
        System.out.println(checkCharacterCase(ch2)); // Lowercase
        System.out.println(checkCharacterCase(ch3)); // Not a letter
        System.out.println(checkCharacterCase(ch4)); // Not a letter
    }

    // Method to check if the character is uppercase, lowercase, or not a letter
    public static String checkCharacterCase(char ch) {
        // Use the ternary operator with ASCII value checks
        return (ch >= 'A' && ch <= 'Z') ? "Uppercase" :
               (ch >= 'a' && ch <= 'z') ? "Lowercase" :
               "Not a letter";
    }
}
```



```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac UpperLowerCase.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java UpperLowerCase
Uppercase
Lowercase
Not a letter
Not a letter
```

Q20: Implement a Java program that returns the absolute value of a given number using the ternary operator (without using Math.abs()).

```
public class AbsoluteValueTernary {
        public static void main(String args[]){
                check(20);
                check(-120);
        }


        public static void check(int A){
                int result = (A>0)? A : -A;
                System.out.println("absolute value: " + result);
        }
}
```



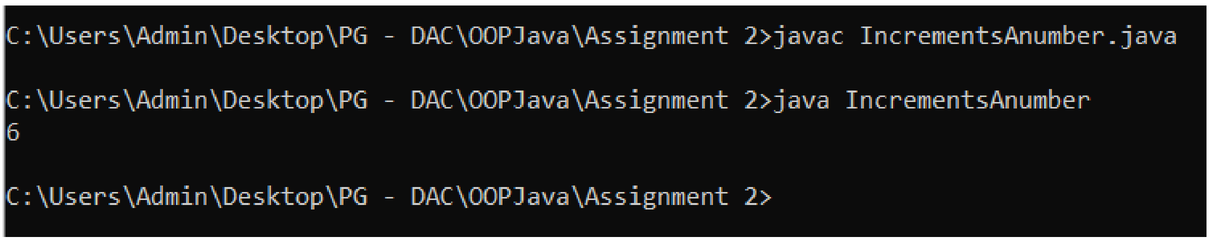
```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac AbsoluteValue.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java AbsoluteValue
absolute value: 20
absolute value: 120

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>
```

5. Miscellaneous Operator Questions

Q21: Write a program that increments a number without using + or ++ operators.

Hint : Use bitwise - (~x) .

```
public class IncrementsAnumber{
        static public void main(String me[]){
                int x = 5;
                System.out.println(-~x);
                // Use bitwise NOT (~) to find the complement and then simulate +1
        }
}
```



```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac IncrementsAnumber.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java IncrementsAnumber
6

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>
```

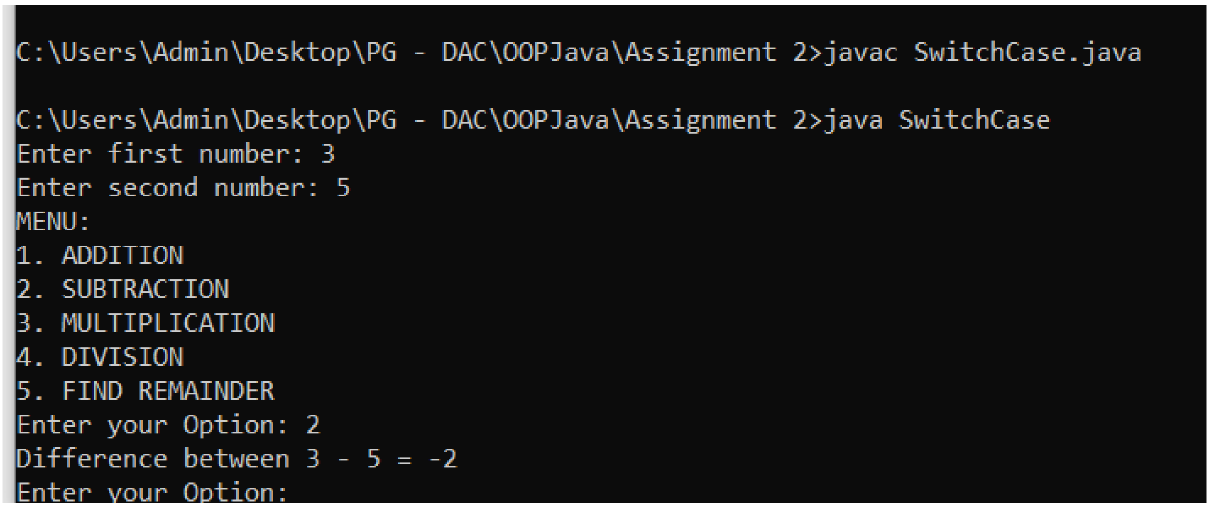
Q22: Implement a calculator that takes two numbers and an operator (+ , - , * , /) as input and prints the result using only switch-case .

```
import java.util.Scanner;

public class SwitchCase   {
   public static void main(String[] args) {
      // Create Scanner object to read input
      Scanner sc = new Scanner(System.in);
```

```
// Prompt the user for the first number
System.out.print("Enter first number: ");
int a = sc.nextInt();

// Prompt the user for the second number
System.out.print("Enter second number: ");
int b = sc.nextInt();

// Display the menu of operations
System.out.println("MENU: ");
System.out.println("1. ADDITION ");
System.out.println("2. SUBTRACTION ");
System.out.println("3. MULTIPLICATION ");
System.out.println("4. DIVISION ");
System.out.println("5. FIND REMAINDER ");
System.out.print("Enter your Option: ");

// Read the user's option
int c = sc.nextInt();

// Loop until the user chooses to exit (option 6)
while (c != 6) {
    // Switch case to perform the selected operation
    switch (c) {
        case 1: // Addition case
            System.out.println("Addition of " + a + " + " + b + " = " + (a + b));
            break;

        case 2: // Subtraction case
```

```
            System.out.println("Difference between " + a + " - " + b + " = " + (a - b));
            break;

          case 3: // Multiplication case
            System.out.println("Multiplication of " + a + " x " + b + " = " + (a * b));
            break;

          case 4: // Division case
            // Perform division with floating point result
            System.out.println("Quotient in division of " + a + " / " + b + " = " + ((float) a /
(float) b));
            break;

          case 5: // Find remainder case
            // Calculate remainder in division
            System.out.println("Remainder in division of " + a + " and " + b + " = " + ((float) a
% (float) b));
            break;

          default: // Case for invalid options
            System.out.println("INVALID OPTION");
            break;
        }

        // Prompt for another option to continue or exit
        System.out.print("Enter your Option: ");
        c = sc.nextInt();
      }

      // Once the user chooses to exit, print an exit message
```

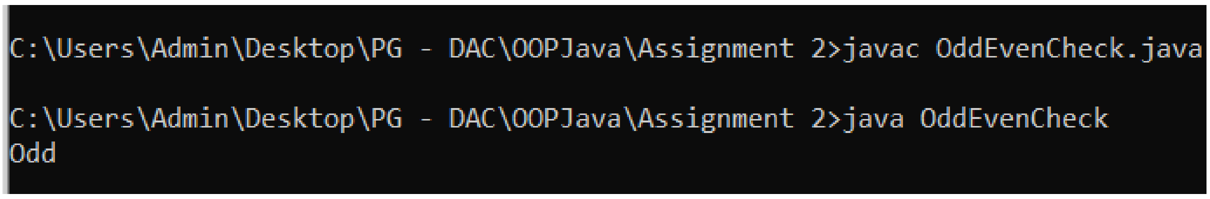
```
        System.out.println("EXITING PROGRAMMING!");
    }
}
```



```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac SwitchCase.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java SwitchCase
Enter first number: 3
Enter second number: 5
MENU:
1. ADDITION
2. SUBTRACTION
3. MULTIPLICATION
4. DIVISION
5. FIND REMAINDER
Enter your Option: 2
Difference between 3 - 5 = -2
Enter your Option:
```

Q23: Given a number, find whether it is odd or even using the & bitwise operator and print the result without using if-else .

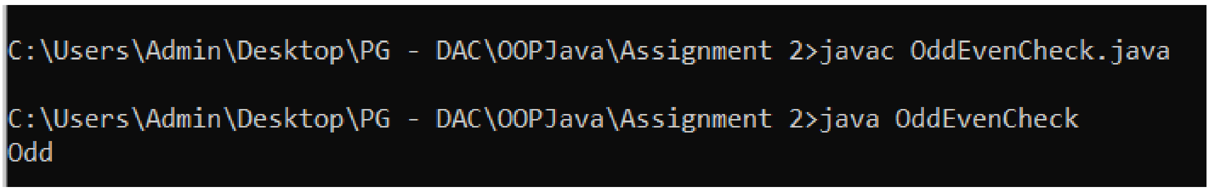
```
public class OddEvenCheck {
    public static void main(String[] args) {
        // Example input
        int num = 7;

        // Using bitwise AND to check if the number is even or odd
        System.out.println((num & 1) == 0 ? "Even" : "Odd");
    }
}
```



```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac OddEvenCheck.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java OddEvenCheck
Odd
```

Q24: Write a program that prints all even numbers from 1 to 100 using only bitwise AND (&) and for loop.

```
public class EvenNumbers {
   public static void main(String[] args) {
      // Use a for loop to iterate from 1 to 100
      for (int num = 1; num <= 100; num++) {
         // Check if the number is even using bitwise AND
         if ((num & 1) == 0) {
            System.out.println(num); // Print the even number
         }
      }
   }
}
```



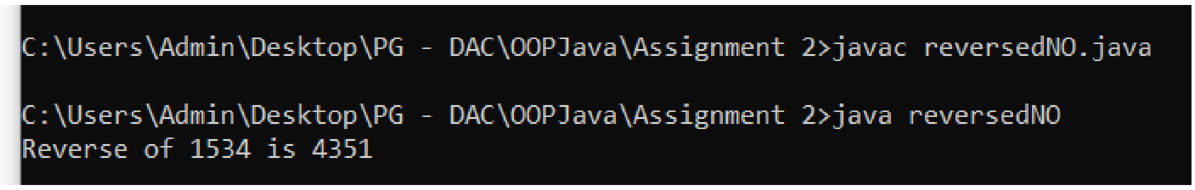
```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac OddEvenCheck.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java OddEvenCheck
Odd
```

Q25: Implement a program that reverses an integer number without using string conversion (StringBuilder or toCharArray).

Hint : Use while(n!=0) { rev = rev * 10 + n % 10; n /= 10; }

```
public class reversedNO {
   // Main method where the program execution begins
   static public void main(String me[]) {
      // Initialize the number to be reversed (x) and a variable to store the reversed number
(rev)
      int x = 1534, rev = 0;
```

```
        // Store the original value of x to use it later in the output (temp)
        int temp = x;

        // Loop until the number x becomes 0
        while(x != 0) {
            // Extract the last digit of x and add it to the reversed number
            rev = rev * 10 + x % 10;

            // Remove the last digit from x by performing integer division by 10
            x = x / 10;
        }

        // Print the result: the original number and its reversed version
        System.out.println("Reverse of " + temp + " is " + rev);
    }
}
```



```
C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>javac reversedNO.java

C:\Users\Admin\Desktop\PG - DAC\OOPJava\Assignment 2>java reversedNO
Reverse of 1534 is 4351
```