Assignment 2

1.Arithmetic & Assignment Operators

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

```
import java.util.Scanner;


public class Swapnoxor {

   public static void main(String[] args) {

     // Scanner scanner = new Scanner(System.in);

      int a = 5;

          int b = 10;

                    System.out.println("Befor Swapping "+"  a= "+a+"  b= "+b);


                    a^=b;

                    b^=a;

                    //a^=b;

                    System.out.println("After Swapping "+"  a= "+a+"  b= "+b);

   }

}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Day 3>javac Swapnoxor.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Day 3>java Swapnoxor
Befor Swapping   a= 5  b= 10
After Swapping   a= 10  b= 5
```

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

```
import java.util.Scanner;

public class EvenorOdd {

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

```
        Scanner sc = new Scanner(System.in);


                        System.out.print("Enter a number: ");
        int num = sc.nextInt();


        System.out.println((num & 1) == 0 ? "Even" : "Odd");
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Day 3>javac EvenorOdd.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Day 3>java EvenorOdd
Enter a number: 19
Odd
```

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

```
import java.util.Scanner;


public class SumofDigit {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);


                        System.out.print("Enter a number: ");
        int num = sc.nextInt();
                        int sum =0;
                        int rem;


                                while (num!=0)
            {
                        rem=num%10;
                        sum=sum+rem;
                        num=num/10;
```

```
        }
        System.out.print("Sum of number: " + sum);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Day 3>javac SumofDigit.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Day 3>java SumofDigit
Enter a number: 1234
Sum of number: 10
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Day 3>
```

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 DivisibilityByThree {
    static boolean isDivisibleBy3(int num) {
        num = Math.abs(num);
        while (num > 0) {
            num -= 3;
        }
        return num == 0;
    }

    public static void main(String[] args) {
        int num = 27;
        System.out.println(num + (isDivisibleBy3(num) ? " is divisible by 3." : " is not divisible by 3."));
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac
 DivisibilityByThree.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java
DivisibilityByThree
27 is divisible by 3.

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>
```

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

```
public class SwapNumbers {
    public static void main(String[] args) {
        int a = 5, b = 10;
        System.out.println("Before swapping: a = " + a + ", b = " + b);

        a += b;
        b = a - b;
        a -= b;

        System.out.println("After swapping: a = " + a + ", b = " + b);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java
SwapNumbers
Before swapping: a = 5, b = 10
After swapping: a = 10, b = 5
```

2. Relational & Logical Operators

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

```
public class LargestOfThree {
    public static void main(String[] args) {
        int a = 10, b = 25, c = 15;

        int largest = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);

        System.out.println("The largest number is: " + largest);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac
 LargestOfThree.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java
LargestOfThree
The largest number is: 25

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>
```

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

```
public class LeapYearCheck {
    public static void main(String[] args) {
        int year = 2024;

        boolean isLeap = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

        System.out.println(year + (isLeap ? " is a leap year." : " is not a leap year."));
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java
LeapYearCheck
2024 is a leap year.

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>
```

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 AtLeastTwoTrue {
    public static void main(String[] args) {
        boolean a = true, b = false, c = true;

        boolean result = (a && b) || (b && c) || (a && c);

        System.out.println("At least two are true: " + result);
```

```
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac
 AtLeastTwoTrue.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java
AtLeastTwoTrue
At least two are true: true
```

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.

```
public class NumberInRange {
    public static void main(String[] args) {
        int num = 35;

        System.out.println(num + " is in range: " + (num >= 20 && num <= 50));
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac
 NumberInRange.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java
NumberInRange
35 is in range: true
```

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

```
public class VowelOrConsonant {
    public static void main(String[] args) {
        char ch = 'e';

        String result = (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
                ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
```

```
                ? "Vowel" : "Consonant";

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

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac
 VowelOrConsonant.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java
VowelOrConsonant
e 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.

```
public class PowerOfTwo {
    public static void main(String[] args) {
        int num = 16;

        boolean isPowerOfTwo = (num > 0) && ((num & (num - 1)) == 0);

        System.out.println(num + " is a power of 2: " + isPowerOfTwo);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac Po
werOfTwo.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Pow
erOfTwo
16 is a power of 2: true
```

Q12: Write a Java program to multiply a number by 8 without using * or / operators. Hint : Use bitwise left shift (<<)

```
public class MultiplyByEight {
    public static void main(String[] args) {
        int num = 5;

        int result = num << 3; // Multiplying by 8 using left shift

        System.out.println(num + " multiplied by 8 is: " + result);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac Mu
ltiplyByEight.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Mul
tiplyByEight
5 multiplied by 8 is: 40
```

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 AbsoluteValue {
    public static void main(String[] args) {
        int num = -10;

        int mask = num >> 31;
        int abs = (num + mask) ^ mask;

        System.out.println("Absolute value of " + num + " is: " + abs);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac Ab
soluteValue.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Abs
oluteValue
Absolute value of -10 is: 10
```

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 CountSetBits {
    public static void main(String[] args) {
        int num = 29; // Example number
        int count = 0;

        while (num > 0) {
            num = num & (num - 1); // Clears the lowest set bit
            count++;
        }

        System.out.println("Number of 1s in binary representation: " + count);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac Co
untSetBits.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Cou
ntSetBits
Number of 1s in binary representation: 4

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\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) {
        int num = 23; // Example number
```

```
        int evenBits = num & 0xAAAAAAAA; // Mask even bits
        int oddBits = num & 0x55555555;  // Mask odd bits

        evenBits >>= 1; // Shift even bits right
        oddBits <<= 1;  // Shift odd bits left

        int result = evenBits | oddBits; // Combine shifted bits

        System.out.println("Number after swapping odd and even bits: " + result);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac Sw
apOddEvenBits.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Swa
pOddEvenBits
Number after swapping odd and even bits: 43
```

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 NumberSignCheck {
    public static void main(String[] args) {
        int num = -10;

        String result = (num > 0) ? "Positive" : (num < 0) ? "Negative" : "Zero";

        System.out.println("The number is: " + result);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac Nu
mberSignCheck.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Num
berSignCheck
The number is: Negative
```

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

```
public class MinOfFour {
    public static void main(String[] args) {
        int a = 10, b = 5, c = 8, d = 3;

        int min = (a < b) ? ((a < c) ? ((a < d) ? a : d) : (c < d ? c : d))
                          : ((b < c) ? ((b < d) ? b : d) : (c < d ? c : d));

        System.out.println("The minimum number is: " + min);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac Mi
nOfFour.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Min
OfFour
The minimum number is: 3
```

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 StudentResult {
    public static void main(String[] args) {
        int percentage = 55; // Example percentage

        String result = (percentage >= 40) ? "Pass" : "Fail";

        System.out.println("Result: " + result);
```

```
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac St
udentResult.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Stu
dentResult
Result: 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 CharacterCheck {
    public static void main(String[] args) {
        char ch = 'A'; // Example character

        String result = (ch >= 'A' && ch <= 'Z') ? "Uppercase" :
                        (ch >= 'a' && ch <= 'z') ? "Lowercase" : "Not a letter";

        System.out.println("Character type: " + result);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac Ch
aracterCheck.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Cha
racterCheck
Character type: Uppercase
```

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) {
        int num = -10; // Example number
```

```
        int absValue = (num < 0) ? -num : num;

        System.out.println("Absolute value: " + absValue);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac Ab
soluteValueTernary.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Abs
oluteValueTernary
Absolute value: 10
```

5. Miscellaneous Operator Questions

Q21: Write a program that increments a number without using + or ++ operators. Hint : Use bitwise -(~x)

```
public class IncrementWithoutPlus {
    public static void main(String[] args) {
        int num = 5; // Example number

        int incremented = -~num; // Using bitwise NOT and negation to increment

        System.out.println("Incremented value: " + incremented);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>javac In
crementWithoutPlus.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Inc
rementWithoutPlus
Incremented value: 6
```

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 Calculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter first number: ");
        double num1 = scanner.nextDouble();

        System.out.print("Enter an operator (+, -, *, /): ");
        char operator = scanner.next().charAt(0);

        System.out.print("Enter second number: ");
        double num2 = scanner.nextDouble();

        double result;

        switch (operator) {
            case '+':
                result = num1 + num2;
                break;
            case '-':
                result = num1 - num2;
                break;
            case '*':
                result = num1 * num2;
                break;
            case '/':
                result = (num2 != 0) ? num1 / num2 : Double.NaN;
                break;
            default:
```

```
            System.out.println("Invalid operator");
            return;
        }

        System.out.println("Result: " + result);
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Cal
culator
Enter first number: 10
Enter an operator (+, -, *, /): +
Enter second number: 20
Result: 30.0

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>
```

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

```
import java.util.Scanner;

public class OddEvenCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int num = scanner.nextInt();

        System.out.println(num + " is " + ((num & 1) == 0 ? "Even" : "Odd"));
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Odd
EvenCheck
Enter a number: 10
10 is Even
```

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

```
public class EvenNumbersBitwise {
    public static void main(String[] args) {
        for (int i = 1; i <= 100; i++) {
            if ((i & 1) == 0) {
                System.out.print(i + " ");
            }
        }
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Eve
nNumbersBitwise
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62
 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>
```

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 ReverseInteger {
    public static void main(String[] args) {
        int num = 12345; // Example number
        int rev = 0;

        while (num != 0) {
            rev = rev * 10 + num % 10;
            num /= 10;
        }

        System.out.println("Reversed number: " + rev);
```

```
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 2>java Rev
erseInteger
Reversed number: 54321
```