JAVA Assignment 2

1. Arithmetic & Assignment Operators

Q1: Swap two numbers using XOR

```
public class SwapNumbersXOR {
   public static void main(String[] args) {
      int a = 5, b = 10;

      a = a ^ b;
      b = a ^ b;
      a = a ^ b;

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

Q2: Check if a number is even or odd using bitwise operator

```
public class EvenOddCheck {
   public static void main(String[] args) {
      int num = 7;

      if ((num & 1) == 0) {
         System.out.println(num + " is even");
      } else {
         System.out.println(num + " is odd");
      }
   }
}
```

Q3: Sum of digits of an integer using % and /

```
public class SumOfDigits {
    public static void main(String[] args) {
        int num = 1234;
        int sum = 0;

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

        System.out.println("Sum of digits: " + sum);
    }
}
```

Q4: Check divisibility by 3 without using % or /

```
public class DivisibilityBy3 {
    public static void main(String[] args) {
        int num = 27;

        while (num > 3) {
            int sum = 0;
            while (num != 0) {
                sum += num & 7; // Extract last 3 bits (approximate modulo 8 behavior)
                num >>= 3; // Equivalent to division by 8
            }
            num = sum;
        }

        if (num == 3 || num == 0) {
            System.out.println("Divisible by 3");
```

```
        } else {
            System.out.println("Not divisible by 3");
        }
    }
}
```

Q5: Swap two numbers using += and -=

```
public class SwapNumbersArithmetic {
    public static void main(String[] args) {
        int a = 5, b = 10;

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

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

---

2. Relational & Logical Operators

```
// Q6: Find the largest of three numbers using 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("Largest number: " + largest);
```

```
    }
}

// Q7: Check if a year is a leap year 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 + " is a leap year: " + isLeap);
    }
}

// Q8: Check if at least two boolean inputs are true
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);
    }
}

// Q9: Check if a number is within range (20 to 50) without if-else
public class NumberInRange {
    public static void main(String[] args) {
        int num = 30;
```

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

// Q10: Check if a character is a vowel or consonant using ternary operator
public class VowelOrConsonant {
    public static void main(String[] args) {
        char ch = 'a';

        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);
    }
}
```

---

3. Bitwise Operators

```
public class BitwiseOperations {

    Q11: Check if a number is a power of 2
    public static boolean isPowerOfTwo(int n) {
        return n > 0 && (n & (n - 1)) == 0;
    }

    Q12: Multiply a number by 8 using bitwise shift
    public static int multiplyByEight(int n) {
        return n << 3; // Left shift by 3 (2^3 = 8)
```

```
}

Q13: Find absolute value using bitwise operators
public static int absoluteValue(int num) {
    int mask = num >> 31; // Get the sign bit
    return (num + mask) ^ mask;
}

Q14: Count number of 1s in binary representation of a number
public static int countSetBits(int n) {
    int count = 0;
    while (n > 0) {
        n &= (n - 1); // Clears the least significant set bit
        count++;
    }
    return count;
}

Q15: Swap odd and even bits
public static int swapOddEvenBits(int x) {
    return ((x & 0xAAAAAAAA) >> 1) | ((x & 0x55555555) << 1);
}

public static void main(String[] args) {
    int num = 16;
    System.out.println("Is " + num + " a power of 2? " + isPowerOfTwo(num));

    int multiplier = 5;
    System.out.println(multiplier + " multiplied by 8 is " +
multiplyByEight(multiplier));
```

```
    int value = -10;
    System.out.println("Absolute value of " + value + " is " + absoluteValue(value));

    int bitCountNum = 29; // Binary: 11101
    System.out.println("Number of set bits in " + bitCountNum + " is " +
countSetBits(bitCountNum));

    int swapNum = 23; // Binary: 10111
    System.out.println("After swapping odd and even bits: " +
swapOddEvenBits(swapNum));
  }
}
```

---

4. Ternary Operator Challenges

```
  // Q16: Return zero using only the ternary operator
  public static int returnZero() {
    return (true ? 0 : 1);
  }

  // Q17: Find the minimum of four numbers using nested ternary operators
  public static int minOfFour(int a, int b, int c, int d) {
    return (a < b ? (a < c ? (a < d ? a : d) : (c < d ? c : d)) : (b < c ? (b < d ? b : d) : (c < d ? c :
d)));
  }

  // Q18: Print "Pass" or "Fail" based on percentage using ternary operator
  public static String passOrFail(int percentage) {
```

```
    return (percentage >= 40 ? "Pass" : "Fail");
  }

  // Q19: Check character case using ternary operator
  public static String checkCharacterType(char ch) {
    return (ch >= 'A' && ch <= 'Z') ? "Uppercase" :
        (ch >= 'a' && ch <= 'z') ? "Lowercase" : "Not a letter";
  }

  // Q20: Return absolute value using ternary operator
  public static int absoluteValueTernary(int num) {
    return num < 0 ? -num : num;
  }

  public static void main(String[] args) {
    int num = 16;
    System.out.println("Is " + num + " a power of 2? " + isPowerOfTwo(num));

    int multiplier = 5;
    System.out.println(multiplier + " multiplied by 8 is " + multiplyByEight(multiplier));

    int value = -10;
    System.out.println("Absolute value of " + value + " is " + absoluteValue(value));

    int bitCountNum = 29; // Binary: 11101
    System.out.println("Number of set bits in " + bitCountNum + " is " +
countSetBits(bitCountNum));

    int swapNum = 23; // Binary: 10111
```

```
    System.out.println("After swapping odd and even bits: " + swapOddEvenBits(swapNum));

    System.out.println("Zero using ternary: " + returnZero());

    System.out.println("Minimum of (12, 5, 20, 8) is " + minOfFour(12, 5, 20, 8));

    int percentage = 35;
    System.out.println("Student result: " + passOrFail(percentage));

    char ch = 'G';
    System.out.println("Character type of '" + ch + "': " + checkCharacterType(ch));

    int absValue = -15;
    System.out.println("Absolute value using ternary: " + absoluteValueTernary(absValue));
  }
}
```

---

5. Miscellaneous Operator Questions

```
// Q21: Increment a number without using + or ++
int increment(int x) {
  return -~x;
}

// Q22: Simple Calculator using switch-case
void calculator(int a, int b, char op) {
  switch(op) {
```

```
    case '+': cout << (a - (-b)) << endl; break; // Using -(-b) instead of +
    case '-': cout << (a + (~b + 1)) << endl; break; // Using Two's complement for subtraction
    case '*': cout << (a * b) << endl; break;
    case '/': if (b != 0) cout << (a / b) << endl; else cout << "Error: Division by zero" << endl; break;
    default: cout << "Invalid operator" << endl;
  }
}

// Q23: Check if a number is even or odd using bitwise &
void checkEvenOdd(int n) {
  cout << ((n & 1) ? "Odd" : "Even") << endl;
}

// Q24: Print all even numbers from 1 to 100 using bitwise & and for loop
void printEvenNumbers() {
  for (int i = 1; i <= 100; i++) {
    if ((i & 1) == 0) { // If LSB is 0, it's even
      cout << i << " ";
    }
  }
  cout << endl;
}

// Q25: Reverse an integer without using string conversion
int reverseNumber(int n) {
  int rev = 0;
  while (n != 0) {
    rev = rev * 10 + (n % 10);
```

```
        n /= 10;
    }
    return rev;
}

int main() {
    // Testing Q21
    cout << "Increment 5: " << increment(5) << endl;

    // Testing Q22
    cout << "Calculator 5 + 3: ";
    calculator(5, 3, '+');

    // Testing Q23
    cout << "Check Even/Odd for 7: ";
    checkEvenOdd(7);

    // Testing Q24
    cout << "Even numbers from 1 to 100: ";
    printEvenNumbers();

    // Testing Q25
    cout << "Reverse of 1234: " << r
```