

JAVA

Assignment no2

Name Akshay janrao

PRN:250240520006

Arithmetic & Assignment Operators

1. 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

Answer The code is as follows

```
public class SwapNumbers {  
  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 20;  
  
        System.out.println("Before swap: a = " + a + ", b = " + b);  
  
        a = a ^ b;  
        b = a ^ b;  
        a = a ^ b;  
  
        System.out.println("After swap: a = " + a + ", b = " + b);  
    }  
}
```

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

Hint: Use n & 1 to check

```
. ublic class EvenOddBitwise {  
  
    public static String isEvenOrOdd(int n) {  
        if ((n & 1) == 0) {  
            return "Even";  
        } else {  
            return "Odd";  
        }  
    }  
}
```

```

public static void main(String[] args) {
    int number1 = 10;
    int number2 = 7;

    System.out.println(number1 + " is " + isEvenOrOdd(number1));
    System.out.println(number2 + " is " + isEvenOrOdd(number2));

}
}

```

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

```

public class SumOfDigits {

    public static int sumDigits(int n) {
        if (n < 0) {
            n = -n;
        }

        int sum = 0;
        while (n > 0) {
            int digit = n % 10;
            sum += digit;
            n /= 10;
        }
        return sum;
    }

    public static void main(String[] args) {
        int number1 = 12345;

        System.out.println("Sum of digits of " + number1 + " is: " + sumDigits(number1));

    }
}

```

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 DivisibleByThree {

    public static boolean isDivisibleBy3(int n) {
        if (n < 0) {
            n = -n; // Handle negative numbers

```

```

    }

    if (n == 0) {
        return true; // 0 is divisible by 3
    }

    if (n < 3) {
        return false; // 1 and 2 are not divisible by 3
    }

    int oddCount = 0;
    int evenCount = 0;

    while (n > 0) {
        if ((n & 1) == 1) {
            oddCount++;
        }
        n = n >> 1;

        if (n > 0 && (n & 1) == 1) {
            evenCount++;
        }
        n = n >> 1;
    }

    return isDivisibleBy3(Math.abs(oddCount - evenCount));
}

public static void main(String[] args) {
    int number1 = 9;
    int number2 = 10;
    int number3 = 27;

    System.out.println(number1 + " is divisible by 3: " + isDivisibleBy3(number1));
    System.out.println(number2 + " is divisible by 3: " + isDivisibleBy3(number2));
    System.out.println(number3 + " is divisible by 3: " + isDivisibleBy3(number3));
}
}

```

5. Write a Java program to swap two numbers using the += and -= operators only
 Answer The code is as followed

```

public class SwapNumbers2 {
    public static void main (String[] args){
        int a = 10;
        int b = 50;
    }
}

```

```

        System.out.println("Before swap a=" + a + ",b =" +b);

        a += b;
        b = a - b;
        a -= b;
        System.out.println("after swapping a =" + a +"b= "+b);

    }

}

```

Write the program to find the largest of three numbers using the ternary operator?

Answer

```

public class Q6 {

    public static void main (String [] args){

        int a = 10;
        int b = 20;
        int c = 50;

        int largest = ( a >= b ) ? ( a >= c ? a : b ) : ( b >= c ? b : c );
        System.out.println("largest no " + largest);
    }
}

```

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

Answer

```

import java.io.*;
import java.util.*;

public class Q7 {
    public static void main(String[] args){
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

        if (n%4 == 0 &&( n%100!=0 | n%400==0))
        {
            System.out.println(n+ "is a leap year.");
        }else
        {
            System.out.println(n+ "is not a leap year.");
        }
    }
}

```

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

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

ANSWER

```
import java.util.Scanner;

public class Q8 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the first boolean value (true/false): ");
        boolean a = scanner.nextBoolean();

        System.out.println("Enter the second boolean value (true/false): ");
        boolean b = scanner.nextBoolean();

        System.out.println("Enter the third boolean value (true/false): ");
        boolean c = scanner.nextBoolean();

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

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

        scanner.close();
    }
}
```

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.

ANSWER

```
import java.util.Scanner;

public class Q9 {
    public static void main(String [] args){
```

```

Scanner scanner = new Scanner(System.in);
System.out.println("enter a no:");
int number = scanner.nextInt();
System.out.println("the no in range of 20 to 50 " +(number>=20 &&
number<= 50));
}
}

```

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

ANSWER

```

import java.util.Scanner;

public class Q10{
    public static void main (String [] args){
        Scanner scanner = new Scanner(System.in);
        System.out.println(" Enter a character");
        char ch = scanner.next().charAt(0);
        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("The character is " + ch + "is a" + result + ".");
    }
}

```

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.

ANSWER

```

import java.util.Scanner;

public class Q11 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

```

```

System.out.print("Enter a positive integer: ");

int n = scanner.nextInt();

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

if (isPowerOfTwo) {
    System.out.println(n + " is a power of 2.");
} else {
    System.out.println(n + " 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 (<<).

```

import java.util.Scanner;

public class Q12 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();

        int result = number << 3;

        System.out.println("The result of multiplying " + number + " by 8 is: " + 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;

```

import java.util.Scanner;

```

```

public class Q13 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();

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

        System.out.println("The absolute value of " + number + " is: " + abs);
    }
}

```

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

Hint: Use $n \& (n - 1)$.

```

import java.util.Scanner;

public class Q14 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

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

        int count = 0;
        while (n != 0) {
            n = n & (n - 1);
            count++;
        }

        System.out.println("The number of 1s in the binary representation is: " + count);
    }
}

```



```
}
```

```
}
```

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

Hint: Use masks: $(x \& 0xAAAAAAAA) \gg 1 \mid (x \& 0x55555555) \ll 1$.

```
import java.util.Scanner;
```

```
public class Q15 {
```

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

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

```
        System.out.print("Enter an integer: ");
```

```
        int number = scanner.nextInt();
```

```
        int oddMask = 0xAAAAAAAA;
```

```
        int evenMask = 0x55555555;
```

```
        int swappedNumber = ((number & oddMask) >> 1) | ((number & evenMask) << 1);
```

```
        System.out.println("The number after swapping odd and even bits is: " + swappedNumber);
```

```
    }}
```

4. Ternary Operator Challenges

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

```
import java.util.Scanner;
```

```
public class Q16 {
```

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

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

```
        System.out.println("Enter a no ");
```

```
            int num = scanner.nextInt();
```

```
            String result = ( num > 0 ) ? "positive" : ( num < 0 ) ? "negative" : "zero";
```

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

```
    }}
```

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

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

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

```
    System.out.print("Enter the first number: ");
```

```
    int num1 = scanner.nextInt();
```

```
    System.out.print("Enter the second number: ");
```

```
    int num2 = scanner.nextInt();
```

```
    System.out.print("Enter the third number: ");
```

```
    int num3 = scanner.nextInt();
```

```
    System.out.print("Enter the fourth number: ");
```

```
    int num4 = scanner.nextInt();
```

```
    int min = (num1 < num2) ? ((num1 < num3) ? ((num1 < num4) ? num1 : num4)
```

```
        : ((num3 < num4) ? num3 : num4) : ((num2 < num3)
```

```
            ? ((num2 < num4) ? num2 : num4)
```

```
        : ((num3 < num4) ? num3 : num4));
```

```
    System.out.println("The minimum of the four numbers is: " + min);
```

```
}  
}
```

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

```
import java.util.Scanner;
```

```
public class q18 {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print("Enter the student's percentage: ");  
        int percentage = scanner.nextInt();  
  
        String result = (percentage >= 40) ? "Pass" : "Fail";  
  
        System.out.println("The student has " + result + "ed.");  
    }  
}
```

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

ANSWER

```
import java.util.Scanner;  
  
public class Q19 {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);
```

```

        System.out.print("Enter a single character: ");

        char ch = scanner.next().charAt(0);

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

        System.out.println("The character '" + ch + "' is: " + result);
    }
}

```

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

```

import java.util.Scanner;

public class Q20 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

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

        int absoluteValue = (number >= 0) ? number : -number;

        System.out.println("The absolute value of " + number + " is: " + absoluteValue);
    }
}

```

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

Hint: Use bitwise - (~x).

```

class Q21 {

```

```

public static void main(String[] args) {
    int number = 5;
    System.out.println("Original number: " + number);

    int incremented = increment(number);

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

```

```

public static int increment(int num) {
    return ~num;
}
}

```

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;
```

```

class Q22 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

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

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

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

        double result;

```

```

switch (operator) {
    case '+':
        result = num1 + num2;
        System.out.println("Result: " + result);
        break;
    case '-':
        result = num1 - num2;
        System.out.println("Result: " + result);
        break;
    case '*':
        result = num1 * num2;
        System.out.println("Result: " + result);
        break;
    case '/':
        if (num2 != 0) {
            result = num1 / num2;
            System.out.println("Result: " + result);
        } else {
            System.out.println("Error: Division by zero is not allowed.");
        }
        break;
    default:
        System.out.println("Error: Invalid operator.");
        break;
}
}
}

```

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

```

class Q23 {

```

```

public static void main(String[] args) {
    int number = 7;

    String result = (number & 1) == 0 ? "Even" : "Odd";

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

```

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

```

class Q24 {
    public static void main(String[] args) {
        System.out.println("Even numbers from 1 to 100 are:");

        for (int num = 1; num <= 100; num++) {
            if ((num & 1) == 0) {
                System.out.print(num + " ");
            }
        }
    }
}

```

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;}

```
class Q25 {  
    public static void main(String[] args) {  
        int number = 12345;  
        System.out.println("Original number: " + number);  
  
        int reversed = reverseNumber(number);  
        System.out.println("Reversed number: " + reversed);  
    }  
  
    public static int reverseNumber(int n) {  
        int rev = 0;  
  
        while (n != 0) {  
            rev = rev * 10 + n % 10;  
            n /= 10;  
        }  
  
        return rev;  
    }  
}
```