**Nishant Patil**

**PG-DAC**

**Java**

**Assignment 2**

**1. Arithmetic & Assignment Operators**

***Q1: Swap two numbers without using a third variable and without using arithmetic operators (+ or -).***

class Pre {

public static void main(String[] args) {

int a = 5, b = 10;

a = a + b;

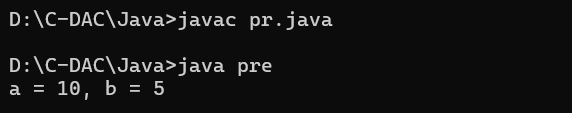
b = a -b;

a = a - b;

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

}

}

****

***Q2: Check whether a given number is even or odd using only bitwise operators.***

class Pre {

public static void main(String[] args) {

int n = 5;

if ((n & 1) == 0) {

System.out.println(n + " is even");

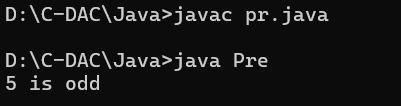
} else {

System.out.println(n + " is odd");

}

}

}

****

***Q3: Calculate the sum of digits of an integer using modulus (%) and division (/) operators.***

class Main {

public static void main(String[] args) {

int n = 1234;

int sum = 0;

while (n > 0) {

sum += n % 10;

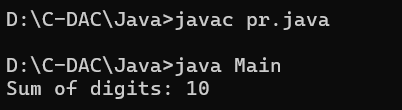
n /= 10;

}

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

}

}

****

**Q4: Check if a number is divisible by 3 without using modulus (%) or division (/).**

class Main {

public static void main(String[] args) {

int n = 15;

while (n >= 3) {

n -= 3;

}

if (n == 0) {

System.out.println("Number is divisible by 3");

}

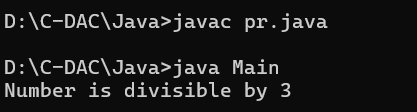
else {

System.out.println("Number is not divisible by 3");

}

}

}

****

***Q5: Swap two numbers using the += and -= operators only.***

class Main {

public static void main(String[] args) {

int a = 5, b = 10;

a += b;

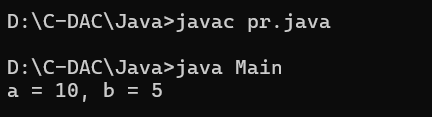
b = a - b;

a -= b;

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

}

}

****

**2. Relational & Logical Operators**

***Q6: Find the largest of three numbers using the ternary operator.***

class Main {

public static void main(String[] args) {

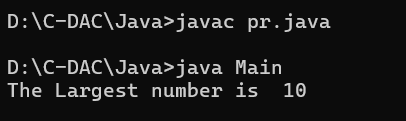
int a = 5, b = 10, c = 8;

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

System.out.println("The Largest number is " + largest);

}

}

****

***Q7: Check whether a given year is a leap year or not using logical (&&, ||) operators.***

class Main {

public static void main(String[] args) {

int year = 2024;

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

System.out.println(year + " is a leap year");

}

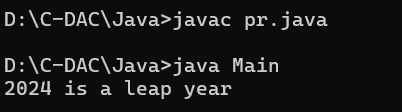
else {

System.out.println(year + " is not a leap year");

}

}

}

****

***Q8: Print true if at least two of three boolean inputs are true.***

class Main {

public static void main(String[] args) {

boolean a = true, b = false, c = true;

if ((a && b) || (a && c) || (b && c)) {

System.out.println("At least two are true");

}

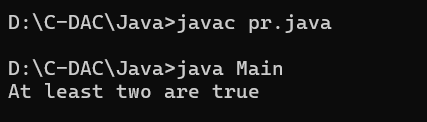
else {

System.out.println("Less than two are true");

}

}

}

****

***Q9: Check if a number is within a specific range (20 to 50) without using if-else.***

class Main {

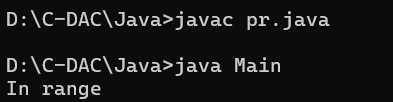
public static void main(String[] args) {

int n= 30;

System.out.println((n >= 20 && n <= 50) ? "In range" : "Out of range");

}

}

****

***Q10: Determine if a character is a vowel or consonant using the ternary operator.***

class Main {

public static void main(String[] args) {

char ch = 'a';

String result = (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||

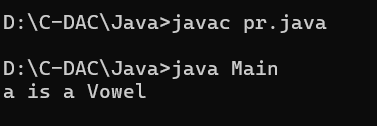
ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') ?

"Vowel" : "Consonant";

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

}

}

****

**3. Bitwise Operators**

***Q11: Check if a given number is a power of 2 using bitwise operators.***

class Main {

public static void main(String[] args) {

int n = 16;

if (n > 0 && (n & (n - 1)) == 0) {

System.out.println(n + " is a power of 2");

}

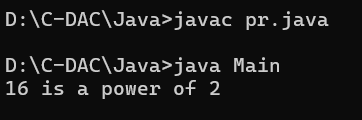
else {

System.out.println(n + " is not a power of 2");

}

}

}

****

***Q12: Multiply a number by 8 without using \* or / operators.***

class Main {

public static void main(String[] args) {

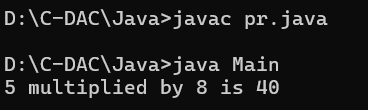
int n = 5;

int result = n << 3;

System.out.println(n + " multiplied by 8 is " + result);

}

}

****

***Q13: Find the absolute value of an integer using bitwise operators.***

class Main {

public static void main(String[] args) {

int num = -5;

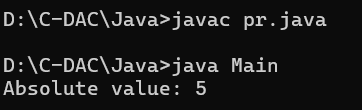
int a = num >> 31;

int abs = (num + a) ^ a;

System.out.println("Absolute value: " + abs);

}

}

****

***Q14: Count the number of 1s (set bits) in the binary representation of a number.***

class Main {

public static void main(String[] args) {

int n = 29;

int count = 0;

while (n > 0) {

n = n & (n - 1);

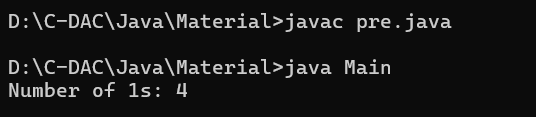
count++;

}

System.out.println("Number of 1s: " + count);

}

}

****

**Q15: Swap odd and even bits of a number using bitwise operators.**

class Main {

public static void main(String[] args) {

int x = 0b1010101010101010;

int even\_bits = x & 0xAAAAAAAA;

int odd\_bits = x & 0x55555555;

even\_bits >>= 1;

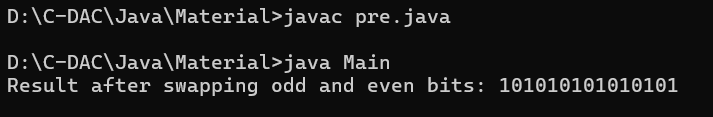
odd\_bits <<= 1;

int result = even\_bits | odd\_bits;

System.out.println("Result after swapping odd and even bits: " + Integer.toBinaryString(result));

}

}

****

**4. Ternary Operator Challenges**

***Q16: Determine whether a given number is positive, negative, or zero using only the ternary operator.***

class Main {

public static void main(String[] args) {

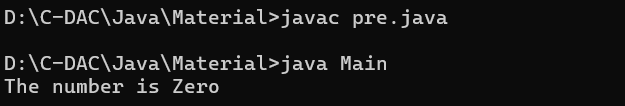
int num = 0;

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

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

}

}

****

***Q17: Find the minimum of four numbers using nested ternary operators.***

class Main {

public static void main(String[] args) {

int a = 5, b = 2, c = 8, d = 6;

int min1 = (a < b) ? a : b ;

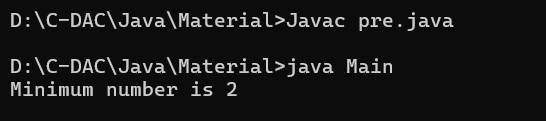
int min2 = (c < d)? c : d ;

int min = ( min1 < min2 )? min1 : min2 ;

System.out.println("Minimum number is " + min);

}

}



***Q18: Print "Pass" if the percentage is 40 or above; otherwise, print "Fail" using only the ternary operator.***

class Main {

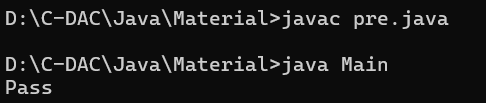
public static void main(String[] args) {

int percentage = 41;

System.out.println((percentage >= 40) ? "Pass" : "Fail");

}

}

****

***Q19: Check if a character is uppercase, lowercase, or not a letter using the ternary operator.***

class Main {

public static void main(String[] args) {

char ch = 'n';

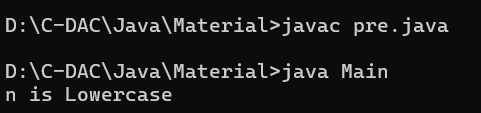
String result = (ch >= 'A' && ch <= 'Z') ? "Uppercase" :

(ch >= 'a' && ch <= 'z') ? "Lowercase" : "Not a letter";

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

}

}

****

***Q20: Return the absolute value of a given number using the ternary operator (without using Math.abs()).***

class Main {

public static void main(String[] args) {

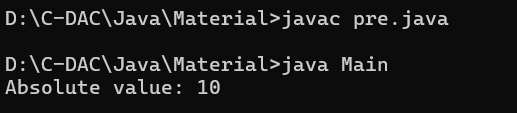
int num = -10;

int absVal = (num < 0) ? -num : num;

System.out.println("Absolute value: " + absVal);

}

}

****

**5. Miscellaneous Operator Questions**

***Q21: Increment a number without using + or ++ operators.***

class Increment {

public static void main(String[] args) {

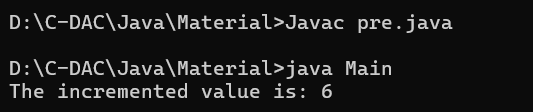
int n = 5;

n = -(~n);

System.out.println("The incremented value is: " + n);

}

}

****

**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 Calculator {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n1 = sc.nextInt();

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

int n2 = sc.nextInt();

System.out.print("Enter the operator (+, -, \*, /): ");

char choice = sc.next().charAt(0);

int result;

switch (choice) {

case '+':

result = n1 + n2;

break;

case '-':

result = n1 - n2;

break;

case '\*':

result = n1 \* n2;

break;

case '/':

if (n2 != 0) {

result = n1 / n2;

} else {

System.out.println("Cannot divide by zero.");

return;

}

break;

default:

System.out.println("Invalid operator.");

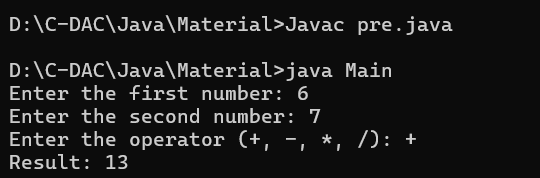
return;

}

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

}

}

****

***Q23: Check whether a number is odd or even using the & bitwise operator without using if-else***

class OddEven {

public static void main(String[] args) {

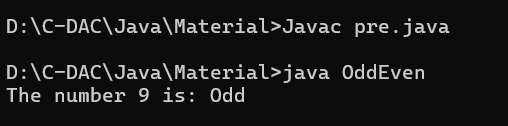
int n = 9;

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

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

}

}

****

***Q24: Print all even numbers from 1 to 100 using only the bitwise AND (&) and a for loop***

class Even {

public static void main(String[] args) {

for (int i = 1; i <= 100; i++) {

if ((i & 1) == 0) {

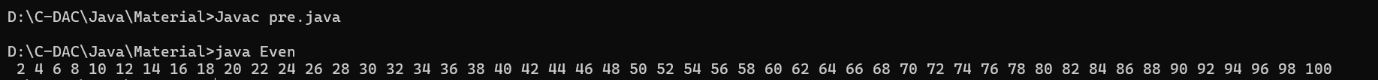
System.out.println(i);

}

}

}

}

****

**Q25: *Reverse an integer number without using string conversion (StringBuilder or toCharArray)***

class Reverse {

public static int reverse(int n) {

int rev = 0;

while (n != 0) {

rev = rev \* 10 + n % 10;

n /= 10;

}

return rev;

}

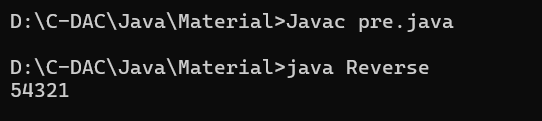
public static void main(String[] args) {

int n = 12345;

System.out.println(reverse(n));

}

}

****