**Assignment-1**

**1. Arithmetic & Assignment Operators**

**1.**

**Declare and Initialize Primitive Data Types**

public class PrimitiveDataTypes {

public static void main(String[] args) {

byte b = 10;

short s = 20;

int i = 30;

long l = 40L;

float f = 50.5f;

double d = 60.6;

char c = 'A';

boolean bool = true;

System.out.println("Byte: " + b);

System.out.println("Short: " + s);

System.out.println("Int: " + i);

System.out.println("Long: " + l);

System.out.println("Float: " + f);

System.out.println("Double: " + d);

System.out.println("Char: " + c);

System.out.println("Boolean: " + bool);

}

}

**2. Convert int to double automatically**

public class IntToDouble {

public static void main(String[] args) {

int num = 10;

double converted = num; // Implicit conversion

System.out.println("Integer value: " + num);

System.out.println("Converted double value: " + converted);

}

}

**3. Convert double to int using typecasting and explain data loss**

public class DoubleToInt {

public static void main(String[] args) {

double num = 10.99;

int converted = (int) num; // Explicit casting

System.out.println("Double value: " + num);

System.out.println("Converted int value (Data loss occurs): " + converted);

}

}

*Data loss occurs because the decimal part (0.99) is truncated when converting to int.*

**4. Calculate the average of three int numbers using typecasting**

public class AverageCalculator {

public static void main(String[] args) {

int a = 10, b = 20, c = 30;

double avg = (a + b + c) / 3.0; // Typecasting

System.out.println("Average: " + avg);

}

}

**5. Demonstrate binary, octal, hexadecimal, and floating-point literals**

public class LiteralsDemo {

public static void main(String[] args) {

int binary = 0b1010; // Binary (10)

int octal = 012; // Octal (10)

int hex = 0xA; // Hexadecimal (10)

float floatNum = 10.5f;

double doubleNum = 20.99;

System.out.println("Binary: " + binary);

System.out.println("Octal: " + octal);

System.out.println("Hexadecimal: " + hex);

System.out.println("Float: " + floatNum);

System.out.println("Double: " + doubleNum);

}

}

**6. Display character and string literals along with their ASCII values**

public class CharStringLiterals {

public static void main(String[] args) {

char ch = 'A';

String str = "Hello";

System.out.println("Character: " + ch + ", ASCII: " + (int) ch);

for (char c : str.toCharArray()) {

System.out.println("Character: " + c + ", ASCII: " + (int) c);

}

}

}

**7. Use boolean literals to control program flow**

public class BooleanFlow {

public static void main(String[] args) {

boolean isRaining = false;

if (isRaining) {

System.out.println("Take an umbrella.");

} else {

System.out.println("Enjoy the sunshine!");

}

}

}

**8. Perform arithmetic operations on two integers**

public class ArithmeticOperations {

public static void main(String[] args) {

int a = 15, b = 4;

System.out.println("Addition: " + (a + b));

System.out.println("Subtraction: " + (a - b));

System.out.println("Multiplication: " + (a \* b));

System.out.println("Division: " + (a / b));

System.out.println("Modulus: " + (a % b));

}

}

**9. Compare two integers using relational operators**

public class RelationalOperators {

public static void main(String[] args) {

int x = 10, y = 20;

System.out.println("x == y: " + (x == y));

System.out.println("x != y: " + (x != y));

System.out.println("x > y: " + (x > y));

System.out.println("x < y: " + (x < y));

System.out.println("x >= y: " + (x >= y));

System.out.println("x <= y: " + (x <= y));

}

}

**10. Check if a number is positive and even using logical operators**

public class LogicalOperators {

public static void main(String[] args) {

int num = 8;

if (num > 0 && num % 2 == 0) {

System.out.println(num + " is positive and even.");

} else {

System.out.println(num + " is not positive or not even.");

}

}

}

**11. Demonstrate the use of assignment operators**

public class AssignmentOperators {

public static void main(String[] args) {

int a = 10, b = 5;

a += b;

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

a -= b;

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

a \*= b;

System.out.println("a \*= b: " + a);

a /= b;

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

a %= b;

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

}

}

**1. Swap Two Numbers Without Third Variable (Using XOR)**

public class SwapXOR {

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);

}

}

**2. Check Even or Odd Using Bitwise AND**

public class EvenOddBitwise {

public static void main(String[] args) {

int num = 7;

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

}

}

**3. Sum of Digits of an Integer**

public class SumOfDigits {

public static void main(String[] args) {

int num = 1234, sum = 0;

while (num > 0) {

sum += num % 10;

num /= 10;

}

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

}

}

**4. Check Divisibility by 3 Without Modulus or Division**

public class DivisibilityBy3 {

public static boolean isDivisibleBy3(int num) {

while (num > 3) {

num = num - 3;

}

return num == 0 || num == 3;

}

public static void main(String[] args) {

int num = 27;

System.out.println(num + " is " + (isDivisibleBy3(num) ? "Divisible by 3" : "Not Divisible by 3"));

}

}

**5. Swap Two Numbers Using += and -=**

public class SwapUsingOperators {

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);

}

}

**6. Largest of Three Numbers Using Ternary Operator**

public class LargestTernary {

public static void main(String[] args) {

int a = 10, b = 20, c = 15;

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

System.out.println("Largest: " + max);

}

}

**7. Check Leap Year Using Logical Operators**

public class LeapYear {

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 " + (isLeap ? "a Leap Year" : "Not a Leap Year"));

}

}

**8. At Least Two Booleans True**

public class AtLeastTwoTrue {

public static void main(String[] args) {

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

System.out.println((a && b) || (b && c) || (a && c));

}

}

**9. Check if Number is in Range (20 to 50)**

public class CheckRange {

public static void main(String[] args) {

int num = 30;

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

}

}

**10. Check if Character is Vowel or Consonant Using Ternary**

public class VowelConsonant {

public static void main(String[] args) {

char ch = 'e';

System.out.println("aeiouAEIOU".indexOf(ch) != -1 ? "Vowel" : "Consonant");

}

}

**11. Check if Number is Power of 2 Using Bitwise Operators**

public class PowerOfTwo {

public static void main(String[] args) {

int n = 8;

System.out.println((n > 0 && (n & (n - 1)) == 0) ? "Power of 2" : "Not a Power of 2");

}

}

**12. Multiply by 8 Using Bitwise Shift**

public class MultiplyBy8 {

public static void main(String[] args) {

int num = 5;

System.out.println("Result: " + (num << 3));

}

}

**13. Absolute Value Using Bitwise Operators**

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: " + abs);

}

}

**14. Count Set Bits in a Number**

public class CountSetBits {

public static void main(String[] args) {

int num = 29, count = 0;

while (num > 0) {

num &= (num - 1);

count++;

}

System.out.println("Set bits: " + count);

}

}

**15. Swap Odd and Even Bits**

public class SwapBits {

public static void main(String[] args) {

int num = 23;

int result = ((num & 0xAAAAAAAA) >> 1) | ((num & 0x55555555) << 1);

System.out.println("After swapping: " + result);

}

}

**16. Determine if Number is Positive, Negative, or Zero**

public class NumberCheck {

public static void main(String[] args) {

int num = -5;

System.out.println(num > 0 ? "Positive" : num < 0 ? "Negative" : "Zero");

}

}

**17. Find Minimum of Four Numbers Using Nested Ternary**

public class MinOfFour {

public static void main(String[] args) {

int a = 10, b = 20, c = 5, d = 15;

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("Minimum: " + min);

}

}

**18. Pass/Fail Using Ternary Operator**

public class PassFail {

public static void main(String[] args) {

int percentage = 35;

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

}

}

**19. Character Type Using Ternary Operator**

public class CharCheck {

public static void main(String[] args) {

char ch = 'A';

System.out.println(ch >= 'A' && ch <= 'Z' ? "Uppercase" :

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

}

}

**20. Increment Number Without Using + or ++**

public class IncrementWithoutPlus {

public static void main(String[] args) {

int num = 5;

System.out.println("Incremented value: " + (-~num));

}

}

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

public class IncrementWithoutPlus {

public static int increment(int x) {

return -~x;

}

public static void main(String[] args) {

int num = 5;

System.out.println("Incremented value of " + num + " is: " + increment(num));

}

}

Q22: Implement a calculator using only switch-case  
  
import java.util.Scanner;

public class CalculatorSwitchCase {

public static void calculator(int a, int b, char operator) {

switch (operator) {

case '+': System.out.println("Result: " + (a - (-b))); break;

case '-': System.out.println("Result: " + (a + (~b + 1))); break;

case '\*': {

int result = 0;

for (int i = 0; i < Math.abs(b); i++) {

result = (b > 0) ? (result - (-a)) : (result - a);

}

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

break;

}

case '/': {

if (b == 0) {

System.out.println("Division by zero is not allowed.");

return;

}

int quotient = 0, sign = (a < 0) ^ (b < 0) ? -1 : 1;

a = Math.abs(a);

b = Math.abs(b);

while (a >= b) {

a -= b;

quotient++;

}

System.out.println("Result: " + (sign \* quotient));

break;

}

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

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int a = scanner.nextInt();

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

int b = scanner.nextInt();

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

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

calculator(a, b, operator);

scanner.close();

}

}

Q23: Check if a number is odd or even using & bitwise operator  
  
public class OddEvenUsingBitwise {

public static void oddEven(int num) {

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

}

public static void main(String[] args) {

int num = 7;

oddEven(num);

}

}

Q24: Print all even numbers from 1 to 100 using only bitwise AND (&)  
  
public class PrintEvenNumbers {

public static void printEvenNumbers() {

for (int i = 2; i <= 100; i += 2) {

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

System.out.print(i + " ");

}

}

System.out.println();

}

public static void main(String[] args) {

System.out.print("Even numbers from 1 to 100: ");

printEvenNumbers();

}

}

Q25: Reverse an integer number without using string conversion  
  
public class ReverseInteger {

public static int reverseNumber(int n) {

int rev = 0;

while (n != 0) {

rev = rev \* 10 + (n % 10);

n /= 10;

}

return rev;

}

public static void main(String[] args) {

int num = 12345;

System.out.println("Reversed number of " + num + " is: " + reverseNumber(num));

}

}