

## Data type & Bitwise Operators:

- 1) WAP to count the number of bits that are set to 1 in an integer.

```
public class CountSetBits {  
    public static int CountSetBits (int num)  
    {  
        int count=0;  
        while (num>0)  
        {  
            count+=num&1;  
            num>>=1;  
        }  
        public static void main (String [] args)  
        {  
            int number=25;  
            int result=CountSetBits (number);  
            System.out.println ("Number of set bits in "+number+" : "+  
                result);  
        }  
    }
```

- 2) Write a java program to count the parity of an integers?

```
public class ParityChecker {  
    public static int checkParity (int num)  
    {  
        int count=0;  
        while (num>0)  
        {  
            count+=num&1;  
            num>>=1;  
        }  
        return count%2==1?1:0;  
    }  
}
```

```
public static void main (String [] args)  
{  
    int number=25;  
    int paritycheck (number);  
}
```

3) public class BitSwapper {

```
public static int swapBITS(int num, int i, int j) {
    if (((num >> i) & 1) != ((num >> j) & 1)) {
        num ^= (1 << i) | (1 << j);
    }
    return num;
}
```

public static void main(String[] args) {

```
int number = 25;
```

```
int i = 2;
```

```
int j = 4;
```

```
System.out.println("Original number:" + Integer.toBinaryString(number));
```

```
int result = swapBITS(number, i, j);
```

```
System.out.println("Number after swapping bits" + i + "and" + j);
```

```
" :" + Integer.toBinaryString(result));
```

```
}
```

```
}
```

4) WAP that takes 64 bit word and returning the 64-bit word consisting of the bits of the input word in reverse order.

public class ReverseBits {

```
public static long reverseBITS(long input) {
```

```
long result = 0;
```

```
for (int i = 0; i < 64; i++) {
```

```
long bit = (input >> i) & 1;
```

```
result |= (bit << (63 - i));
```

```
}
```

```
return result; }
```

```
public static void main (String [] args) {  
    long inputWord = 161010101010101010101010L;  
  
    System.out.println ("Original Word: " + Long.toBinaryString  
        (long.reverseWord = reverseBits (input  
            word)));  
  
    System.out.println ("Reversed Word: " + Long.toBinary  
        String (reverse  
            -word));  
}
```

5) WAP to compute  $x * y$  without arithmetic operators.

```
public class Multiply {  
    public static int multiply (int x, int y) {  
        int result = 0;  
        while (y != 0) {  
            if ((y & 1) == 1) {  
                result = add (result, x);  
            }  
            y >>= 1;  
            x <<= 1; }  
        return result;  
    }
```

```
    public static int add (int a, int b) {  
        while (b != 0) {  
            int carry = a & b;  
            a = a ^ b;  
            b = carry << 1; }  
        return a; }
```

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

```
int x=5;
int y=7;
int result=multiply(x,y);
System.out.println("Product of "+x+" and "+y+" is "+result);
}
```

6) Write to compute  $x/y$  without arithmetic operators.

```
public class Divide {
    public static int divide(int x,int y) {
        if (y==0) {
            throw new ArithmeticException("Division by zero
is undefined.");
        }
        int quotient=0;
        while (x>=y) {
            x=subtract(x,y);
            quotient++;
        }
        return quotient;
    }

    public static int subtract(int a,int b) {
        while (b!=0) {
            int borrow=(~a)&b;
            a=a^b;
            b=borrow<<1;
        }
        return a;
    }
}
```

```
public static void main(String[] args) {
    int x = 21;
    int y = 3;
    int result = divide(x, y);
    System.out.println("Division of " + x + " by " + y + ":" + result);
}
```

7) Write a program to find  $x^y$ .

```
public class PowerCalculation {
```

```
public static long powerWithLoop(int x, int y) {
    long result = 1;
    for (int i = 0; i < y; i++) {
        result *= x;
    }
    return result;
}
```

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

```
    int x = 2;
```

```
    int y = 5;
```

```
    long resultWithLoop = powerWithLoop(x, y);
```

```
    System.out.println(x + " raised to the power " + y + ":" +
```

```
    resultWithLoop);
```

8) WAP to find the reverse of a number.

```
public class ReverseNumber {
```

```
public static int reverse(int num) {
```

```
    int reverseNum = 0;
```

```
    while (num != 0) {
```

```
        int digit = num % 10;
```

```
        reversedNum = reverseNum * 10 + digit;
```

```
num /= 10;  
}  
public static void main(String[] args) {  
    int inputNumber1 = 123; int reversedNumber1  
    System.out.println("Reverse of " + inputNumber1 +  
        " is " + reversedNumber1);
```

9) WAP to check whether a number is palindrome or not.

```
public class PalindromeCheck {  
    public static boolean isPalindrome(int num) {  
        int originalNum = num;  
        int reverseNum = 0;  
        while (num > 0) {  
            int digit = num % 10;  
            reverseNum = reverseNum * 10 + digit;  
            num /= 10;  
        }  
        return originalNum == reverseNum;  
    }
```

```
public static void main(String[] args) {  
    int palindromeNumber = 121;  
    int isPalindrome1 = isPalindrome(palindromeNumber);
```

```
    System.out.println(palindromeNumber + " is a palindrome: " +  
        isPalindrome1);  
}
```

10) WAP that reads an integer number & counts the number  
of digits are even.

```
public class CountEvenDigits {  
    public static int countEvenDigits(int num) {  
        int count = 0;
```

```
13) While (num != 0) {  
    int digit = num % 10;  
    If (digit % 2 == 0) {  
        count++;  
    }  
    num /= 10;  
}  
Return count;  
}  
Public static void main (String args) {  
    int evenDigitsCount = countEvenDigits (12345678);  
    System.out.println ("Number of even digits is " + evenDigits  
    count);  
}
```

12) public class CreateThirdNum {  
 public static int createThirdNum (int num1, int num2) {  
 int firstTwoDigits = num1 / 100;  
 int lastTwoDigits = num2 % 100;  
 return firstTwoDigits \* 100 + lastTwoDigits;  
 }  
 public static void main (String args) {  
 int num1 = 45678;  
 int result = createThirdNumber (num1);  
 System.out.println ("Third Number created " + result);  
 }  
}

13) WAP to count the frequency of each digit of a number.

public class DigitFrequency {

    public static void countDigitFrequency(int num) {

        int[] frequency = new int[10];

        while (num != 0) {

            int digit = num % 10;

            frequency[digit]++;

        num /= 10;

    for (int i = 0; i < 10; i++) {

        if (frequency[i] > 0) {

            System.out.println("Frequency of digit " + i + ":"

                + frequency[i]);

    }

    public static void main(String[] args) {

        int inputNumber = 12233344;

        System.out.println("Digit frequencies for the number "

            + inputNumber + ":");

        CountDigitFrequency(inputNumber)

}

14) WAP to check whether a number is prime or not.

public class PrimeChecker {

    public static boolean isPrime(int num) {

        if (num <= 1) {

            return false;

}

```

for (int i=2; i<=Math.sqrt(num); i++) {
    if (num % i == 0)
        return false;
}
return true;
}

```

```

public static void main (String [] args) {
    int num = 17;
    if (isPrime (num)) {
        System.out.println (num + " is a prime number");
    } else {
        System.out.println (num + " is not a prime number");
    }
}

```

- 16) Write a Java program to print the prime numbers in range.

```

public class PrimeInRange {
    public static boolean isPrime (int num) {
        if (num <= 1) {
            return false;
        }
        for (int i=2; i<=Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }
}

```

```

public static void pointPrimeInRange(int start,
                                     int end) {
    System.out.println("Prime numbers in the range
                        [ " + start + ", " + end + " ] : ");
    for (int i = start; i <= end; i++) {
        if (isPrime(i)) {
            System.out.println(i + " ");
        }
    }
    System.out.println();
}

public static void main(String[] args) {
    int rangeStart = 10;
    int rangeEnd = 50;
    pointPrimeInRange(rangeStart, rangeEnd);
}

```

- 17) Write a program that returns true if the number is even else return false. Note : Don't use if else.

## Wrapper Class :

- 1) WAP to convert an integer to an integer object.

```
public class IntToIntegerConverter {
```

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

```
        int intValue = 42;
```

```
        Integer integerValue = Integer.valueOf(intValue);
```

```
        System.out.println("Converted Integer Object: " + integerValue);
```

- 2) WAP to convert a float to float object.

```
public class Float {
```

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

```
        float floatValue = 3.14f;
```

```
        float floatObject = Float.valueOf(floatValue);
```

```
        System.out.println("Converted float object: " + floatObject);
```

- 3) WAP to convert a double to double object:

```
public class Double {
```

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

```
        double doubleValue = 3.1419;
```

```
        Double doubleObj = Double.valueOf(doubleValue);
```

```
        System.out.println("Converted Double object: " + doubleObj);
```

- 4) WAP to convert a boolean to Boolean object:

```
public class Boolean {
```

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

```
        boolean booleanVal = true;
```

```
        Boolean booleanObj = Boolean.valueOf(booleanVal);
```

```
        System.out.println("Converted Boolean value: " + booleanObj);
```

```
    }
```

- 5) Write to read an integer as a string and convert it to an Integer object.

- 6) Write a program to read a floating-point number from the user and convert it to a float object.

```
import java.util.Scanner;  
  
public class StringtofloatObjectConverter {  
    public static void main (String [] args) {  
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter a float as a string:");  
        String input = scanner.nextLine();  
        try {  
            float float_value = Float.valueOf (input);  
            System.out.println ("Converted float object: " + float_value);  
        } catch (NumberFormatException e) {  
            System.out.println ("Invalid input, please enter a valid  
                float string");  
        }  
    }  
}
```

- 7) WAP to read a double value as string and convert it to double object.

```
import java.util.Scanner;  
public class StringtoDouble {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter a double value as String");  
        String input = sc.nextLine();  
        try {  
            Double val = Double.parseDouble(input);  
            System.out.println("Converted double value: " + val);  
        } catch (NumberFormatException e) {  
            System.out.println("Invalid input");  
        }  
    }  
}
```

- 8) WAP to read a boolean value as string and convert it to the Boolean object.

```
import java.util.Scanner;  
public class StringtoBoolean {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter a Boolean value as String");  
        boolean input = sc.nextBoolean();  
        try {  
            Boolean val = Boolean.valueOf(input);  
            System.out.println("Converted boolean value: " + val);  
        } catch (NumberFormatException e) {  
            System.out.println("Invalid input");  
        }  
    }  
}
```

10) Write a program to design a simple calculator. The calculator works as follows. input: "123 + 345" output: sum = 468

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

```
public class SimpleCalc {
```

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

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

```
        System.out.println("Enter a mathematical  
expression:");
```

```
        String expression = sc.nextLine();
```

```
        String[] parts = expression.split("[+\\-\\*\\/]");
```

```
try {
```

```
    double operand1 = Double.parseDouble(parts[0]);
```

```
    double operand2 = Double.parseDouble(parts[1]);
```

```
    char operator = expression.charAt(parts[0].length());
```

```
    double result = 0;
```

```
    switch(operator) {
```

```
        case '+':
```

```
            System.out.println("Sum" + (operand1 + operand2));  
            break;
```

```
        case '-':
```

```
            System.out.println("Difference" + (operand1 - operand2));  
            break;
```

```
        case '*':
```

```
            System.out.println("Multiplication" + (operand1 * operand2));  
            break;
```

```
        case '/':
```

```
            if(operand2 != 0) {  
                System.out.println("Quotient" + (operand1 / operand2));  
            }  
            else
```

```
                System.out.println("Error");  
            break;
```

```
        default:
```

```
            System.out.println("Error");  
    }
```

11) WAP that reads a double number as string and converts it to a double base type.

```
public class StringToDouble {  
    public static void main (String [] args) {  
        String doubleString = "3.14";  
        try {  
            Double doubleObj = Double.parseDouble (doubleString);  
            double doubleValue = doubleObj.doubleValue ();  
            System.out.println ("converted double value: " +  
                doubleValue);  
        }  
        catch (NumberFormatException e) {  
            System.out.println ("Invalid input");  
        }  
    }  
}
```

12) WAP that reads an integer as string and converts it to an int base type.

```
public class StringToIntegers {  
    public static void main (String [] args) {  
        String intString = "2";  
        try {  
            Integer intObj = Integer.parseInt (intString);  
            int intValue = intObj.intValue ();  
            System.out.println ("converted int value: " +  
                intValue);  
        }  
        catch (NumberFormatException e) {  
            System.out.println ("Invalid input");  
        }  
    }  
}
```