Basic Programs

Hello World

For beginners actually start with this code. That’s why I am start with this and latest version too short for understanding everyone.

```
class HelloWorld {
    public static void main(String[] args)
    {
        System.out.println("Hello, World!");
    }
}

PS C:\ayekiran> cd JavaSE\core\Chapter01\Programs\normal\basic
Programs\normal\basic> javac HelloWorld.java
Programs\normal\basic> java HelloWorld
Hello, World!
```

Take input from USER

If the user is given some information at runtime, solve the problem at the instant of time.

```
import java.util.Scanner;

class Wish {
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter yourname:  ");     //Kiran
        String userName = input.next();

        System.out.printf("Hello, %s\n", userName); //Hello, Kiran
    }
}
```

Two Number Sum

If take two number a, b. Here printf is suitable because starting use string after all default consider string.

System.out.println(“Sum: “ + firstNumber + secondNumber)   // Sum:1020

Slove this problem using System.out.printf(“Sum: %d”, firstNumber + secondNumber). Shown answer below:

```
public class TwoNumberSum {
    public static void main(String[] args)
    {
        int firstNumber = 10;
        int secondNumber = 20;

        System.out.printf("Sum: %d", firstNumber + secondNumber);
    }  // Sum: 30
}
```

Find Even /Odd

To find even or odd that number is completely divisible by 2 is EVEN, otherwise ODD number.

```
public class EvenOdd {
    public static void main(String[] args)
    {
        int number = 10;

        if(number%2 == 0)
            System.out.println("Even"); //Even
        else
            System.out.println("Odd");
    }
}
```

Factorial of a Number

Factoria number is number that multiply one to itself. For example: 5! = 1× 2 × 3 × 4 × 5. Here logic is number multiply with 1 to n. let's see in the code:

```
public class FactorialNumber {

    public static void main(String[] args) {

        int number = 5;
        int factorial = 1;

        for(int i = 1; i <= number; i++) {
            factorial *= i;
        }

        System.out.printf("%d factorial is = %d", number, factorial);
    }   // 5 factorial is = 120
}
```

Fibonacci Series

Fibonacci series is a series of taken to values a, b third number get from sum a, b like that series will continue. Here see a example: a = 0   b = 1   c = a+b = 1   d = c + b = 2   e = d + c = 3 … so, series is 0   1   1   2   3   5   8   13   21 …

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

        int firstNumber = 0;
        int secondNumber = 1;
        int numberOfIterations = 10;

        System.out.printf("%d  ", firstNumber);
        System.out.printf("%d  ", secondNumber);

        for(int i = 2; i < numberOfIterations; i++) {
            int nextNumber = firstNumber + secondNumber;

            firstNumber = secondNumber;
            secondNumber = nextNumber;

            System.out.printf("%d  ", nextNumber);
        } // 0 1 1 2 3 5 8 13 21 34
    }
}
```

Palindrome Number

A number is remains same when digits are reversed. For example: 101, 2, 5, 1001, 111 … Here logic is originalNumber modulus by 10 that reversedNumber = *10 + rem number.

```
public class PalindromeNumber {
    public static void main(String[] args) {
        int originalNumber = 1;
        int reversedNumber = 0;
        int temp = originalNumber;

        for(; temp != 0; temp /= 10) {
            int digit = temp % 10;
            reversedNumber = reversedNumber * 10 + digit;
        }

        if(reversedNumber == originalNumber)
            System.out.println("Palindrome");
        else
        System.out.println("Not Palindrome");
    } // Palindrome
}
```

Armstrong Number

A number is equal to the sum of its own digits, each raised to the power of the number of digits in the number. For example: 1, 2, 153, 370, 407 ….

Logic, first find length of the number, raised power each digit and add. It will return same.

```
public class ArmstrongNumber {
    public static void main(String[] args) {
        int number = 153;
        int originalNumber = number;
        int sum = 0;
        int digits = 0;

        while (number != 0) {
            number /= 10;
            digits++;
        }

        number = originalNumber;
        for(; number != 0; number/=10) {
            int digit = number % 10;
            sum += Math.pow(digit, digits);
        }

        if(sum == originalNumber)
            System.out.println("Armstrong Number");
        else
            System.out.println("Not Armstrong Number");
    } //Armstrong Number
}
```

Prime Number Check

A number only divisible by 1 and itself. Here logic is only two factors.

```
Public class PrimeNumberCheck {
    public static void main(String[] args) {
        int number = 97;
        int count = 0;

        for (int I = 1; I <= number; i++) {
            if (number % I == 0) {
                count += 1;
            }
        }

        if(count == 2)
            System.out.println(“Prime”);
        else
            System.out.println(“Composite”);
    }
}
```

Perfect Number

A number is equal to the sum of its factors without consideration of itself. For example: 6 = 1 + 2 + 3 , 28.

```
public class PerfectNumber {
    public static void main(String[] args) {
        int number = 28;
        int sum = 0;
        int temp = number;

        for (int i = 1; i < number; i++) {
            if (number % i == 0) {
                sum += i;
            }
        }

        if(sum == temp)
            System.out.println("Perfect");
        else
            System.out.println("NOT Perfect");
    } //Perfect
}
```