



UNIVERSITI
MALAYSIA
PERLIS

NMK30703: Programming for Networks

Lab module 1:

Introduction to JAVA

Appendix 1

Mohamed Elshaikh

Faculty of Electronics Engineering Technology – UniMAP (FTKEN-UniMAP)

Examples with Solutions

1. PrintLeapYears (Given Example)

Problem: Print all leap years between AD 999 and AD 2010 and count them.

Solution:

```
public class PrintLeapYears {
    public static void main(String[] args) {
        int count = 0;
        for (int year = 999; year <= 2010; year++) {
            if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
                System.out.println(year);
                count++;
            }
        }
        System.out.println("Total leap years: " + count);
    }
}
```

2. PrintEvenNumbers

Problem: Print all even numbers between 1 and 100 and count them.

Solution:

```
public class PrintEvenNumbers {
    public static void main(String[] args) {
        int count = 0;
        for (int i = 1; i <= 100; i++) {
            if (i % 2 == 0) {
                System.out.println(i);
                count++;
            }
        }
        System.out.println("Total even numbers: " + count);
    }
}
```

3. SumOfDigits

Problem: Calculate the sum of digits of a given number (e.g., $1234 \rightarrow 1 + 2 + 3 + 4 = 10$).

Solution:

```
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. FactorialCalculator

Problem: Compute the factorial of a given number (e.g., $5! = 120$).

Solution:

```
public class FactorialCalculator {
    public static void main(String[] args) {
        int n = 5, factorial = 1;
        for (int i = 1; i <= n; i++) {
            factorial *= i;
        }
        System.out.println(n + "! = " + factorial);
    }
}
```

5. PrimeNumberChecker

Problem: Check if a given number is prime (e.g., $7 \rightarrow$ Prime, $8 \rightarrow$ Not Prime).

Solution:

```
public class PrimeNumberChecker {
    public static void main(String[] args) {
        int num = 7;
        boolean isPrime = true;
```

```
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                isPrime = false;
                break;
            }
        }
        System.out.println(num + (isPrime ? " is Prime" : " is Not Prime"))
    }
}
```

Lab Exercises

1. PrintOddNumbers

Problem: Print all odd numbers between 50 and 100 and count them.

2. ReverseNumber

Problem: Reverse a given number (e.g., 1234 → 4321).

3. FibonacciSeries

Problem: Print the first n terms of the Fibonacci series (e.g., $n=5 \rightarrow 0, 1, 1, 2, 3$).

4. PalindromeChecker

Problem: Check if a given number is a palindrome (e.g., 121 → Palindrome, 123 → Not Palindrome).

5. ArmstrongNumberChecker

Problem: Check if a given number is an Armstrong number (e.g., 153 → Armstrong, 123 → Not Armstrong).