

WEEK 1 (REMAINING PROGRAMS)

```
import java.util.Scanner;

public class LargestNumber {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Input three numbers
        System.out.println(x: "Enter first number:");
        int num1 = sc.nextInt();

        System.out.println(x: "Enter second number:");
        int num2 = sc.nextInt();

        System.out.println(x: "Enter third number:");
        int num3 = sc.nextInt();

        // Find the largest number
        int largest;
```

```
        // Find the largest number
        int largest;

        if (num1 >= num2 && num1 >= num3) {
            largest = num1;
        } else if (num2 >= num1 && num2 >= num3) {
            largest = num2;
        } else {
            largest = num3;
        }

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

        sc.close();
    }
}
```

```
PS C:\Users\VLT-BNT\OneDrive\Documents\Desktop\OOP JAVA> java LargestNumber
Enter first number:
23
Enter second number:
22
Enter third number:
45
The largest number is: 45
```

```

import java.util.Scanner;

public class SimpleCalculator {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Input two numbers
        System.out.println(x: "Enter first number:");
        double num1 = sc.nextDouble();

        System.out.println(x: "Enter second number:");
        double num2 = sc.nextDouble();

        // Input the operation
        System.out.println(x: "Choose operation (+, -, *, /):");
        char operator = sc.next().charAt(index: 0);

        double result;

        // Perform calculation using switch
        switch (operator) {
            case '+':
                result = num1 + num2;
                break;

            case '-':
                result = num1 - num2;
                break;

            case '*':
                result = num1 * num2;
                break;

            case '/':
                if (num2 == 0) {
                    System.out.println(x: "Error! Division by zero.");
                    sc.close();
                    return;
                }
                result = num1 / num2;
                break;

            default:
                System.out.println(x: "Invalid operator!");
                sc.close();
                return;
        }

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

        sc.close();
    }
}

```

```

PS C:\Users\VLT-BNT\OneDrive\Documents\Desktop\OOP JAVA> java SimpleCalculator
Enter first number:
43
Enter second number:
23
Choose operation (+, -, *, /):
*
Result: 989.0

```

```
import java.util.Scanner;

public class SumOfTwoPrimes {

    // Method to check if a number is prime
    public static boolean isPrime(int n) {
        if (n <= 1) return false;
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) return false;
        }
        return true;
    }
}
```

```
Run | Debug | Run main | Debug main
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.println("Enter a number:");
    int number = sc.nextInt();
    boolean found = false;

    // Check all pairs of numbers less than number
    for (int i = 2; i <= number / 2; i++) {
        int j = number - i;
        if (isPrime(i) && isPrime(j)) {
            System.out.println(number + " can be expressed as the sum of two primes: " + i + " + " + j);
            found = true;
            break; // comment this if you want all possible pairs
        }
    }

    if (!found) {
        System.out.println(number + " cannot be expressed as the sum of two prime numbers.");
    }

    sc.close();
}
```

```
PS C:\Users\VLT-BNT\OneDrive\Documents\Desktop\OOP JAVA> java SumOfTwoPrimes
Enter a number:
33
33 can be expressed as the sum of two primes: 2 + 31
```

```

import java.util.Scanner;

public class PalindromeCheck {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println(x: "Enter a number:");
        int num = sc.nextInt();
        int originalNum = num;
        int reversed = 0;

        // Reverse the number
        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }

        // Check palindrome
        if (originalNum == reversed) {
            System.out.println(originalNum + " is a palindrome number.");
        } else {
            System.out.println(originalNum + " is not a palindrome number.");
        }

        sc.close();
    }
}

```

```

PS C:\Users\VLT-BNT\OneDrive\Documents\Desktop\OOP JAVA> java PalindromeCheck
Enter a number:
23
23 is not a palindrome number.
PS C:\Users\VLT-BNT\OneDrive\Documents\Desktop\OOP JAVA> java PalindromeCheck
Enter a number:
22
22 is a palindrome number.

```

```

import java.util.Scanner;

public class QuadraticEquation {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Input coefficients
        System.out.println(x: "Enter coefficient a:");
        double a = sc.nextDouble();

        System.out.println(x: "Enter coefficient b:");
        double b = sc.nextDouble();

        System.out.println(x: "Enter coefficient c:");
        double c = sc.nextDouble();

        // Check if a is zero
        if (a == 0) {
            System.out.println(x: "This is not a quadratic equation.");
            sc.close();
            return;
        }

        // Calculate discriminant
        double discriminant = b * b - 4 * a * c;

```

```

        if (discriminant > 0) {
            // Two distinct real roots
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println(x: "Two real solutions:");
            System.out.println("Root 1 = " + root1);
            System.out.println("Root 2 = " + root2);
        } else if (discriminant == 0) {
            // One real root
            double root = -b / (2 * a);
            System.out.println(x: "One real solution:");
            System.out.println("Root = " + root);
        } else {
            // Discriminant < 0
            System.out.println(x: "No real solutions exist.");
        }

        sc.close();
    }
}

```

```

PS C:\Users\VLT-BNT\OneDrive\Documents\Desktop\OOP JAVA> java QuadraticEquation
Enter coefficient a:
-2
Enter coefficient b:
43
Enter coefficient c:
-5
Two real solutions:
Root 1 = 0.1169148409316314
Root 2 = 21.38308515906837

```