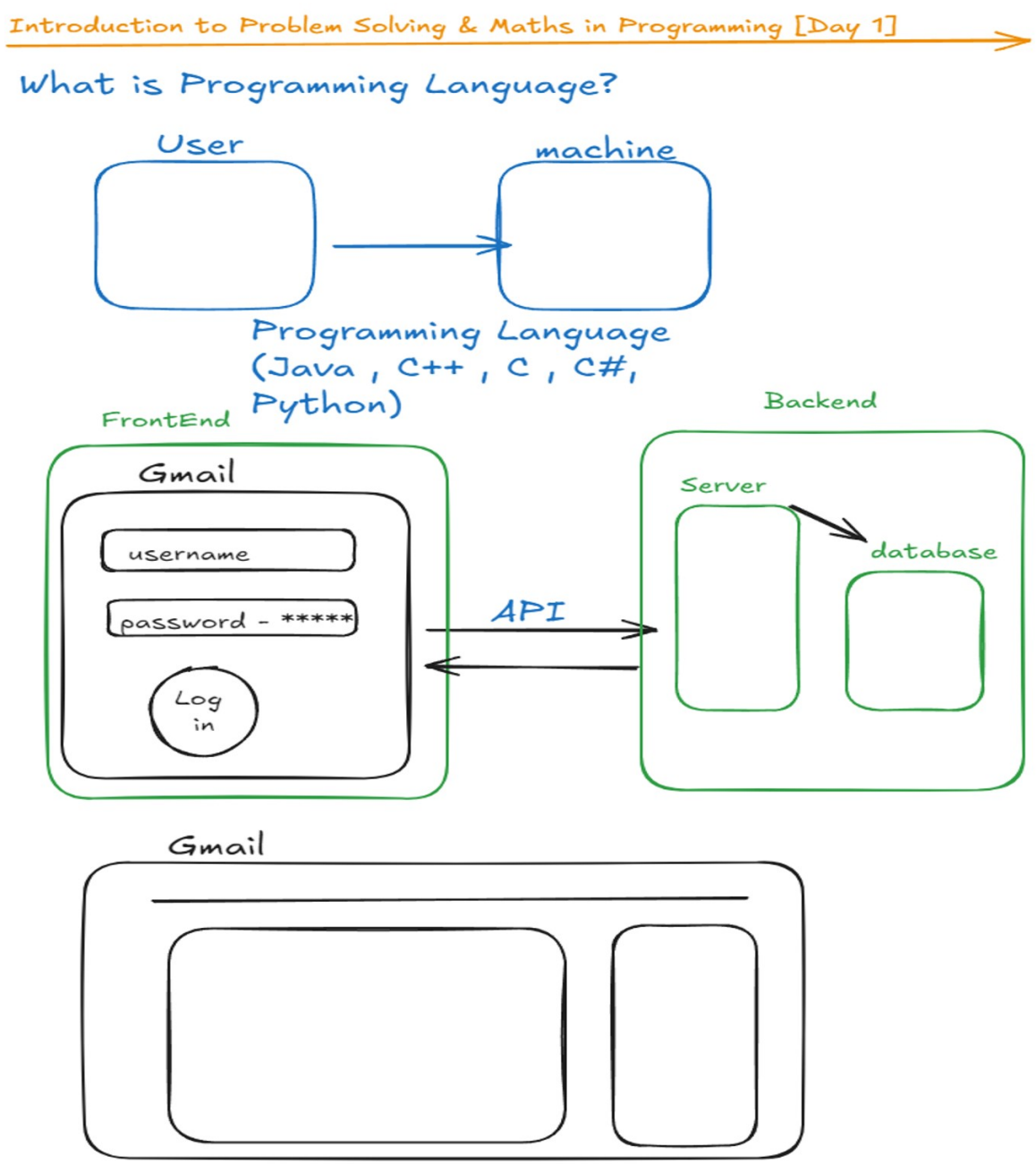
Introduction to Problem Solving & Maths in Programming [Day 1]

02 September 2024 16:02



What is an algorithm

Adding 2 numbers entered by the user.

```
Step1 : Start
Step2 :Declare variables num1 , num2 , sum.
Step3 :Read value num1 & num2.
Step4 :sum = num1+num2
Step5 :Display the sum.
Step6 : Stop
```

Qualities of a Good Algorithm

- Input and output should be defined precisely.
- Each step in the algorithm should be clear and unambiguous.
- Algorithms should be most effective among many different ways to solve a problem.
- An algorithm shouldn't include computer code. Instead, the algorithm should be written in such a way that it can be used in different programming languages.

Write an algorithm to find Largest number among 3 numbers.

Write an algorithm to find factorial of a number.

5! = 5 * 4 * 3 * 2 * 1 = 120.

Write an algorithm to find factorial of a number.

5! = 5 * 4 * 3 * 2 * 1 = 120.
factorial = 1;

```
Step1 : Start
Step 2: Declare variables n , fact , i.
Step 3: Initialize variables :
            fact = 1;
            i = 1;
Step 4:Read the value of n;
Step 5: Repeat the steps until i ==n;
        5.1 : fact *= i;
        5.2 = i+i+1;
Step 6 : Display fact;
Step 7 : Stop
```

```
fact = 720;
```

Memory

```
fact = 720;
i = 7;
n = 6
```

→ Check Whether a number is prime or not

24 ~4.8

1 * 24	24 * 1
2 * 12	12 * 2
3 * 8	8 * 3
4 * 6	6 * 4

36

1 * 36	36 * 1
2 * 18	18 * 2
3 * 12	12 * 3
4 * 9	9* 4

6*6

What are Data Structures?

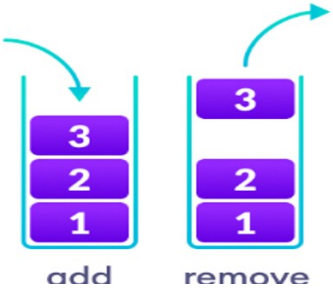
Data structure is a storage that is used to store and organize data. It is a way of arranging data on a computer so that it can be accessed and updated efficiently.

Depending on your requirement and project, it is important to choose the right data structure for your project. For example, if you want to store data sequentially in the memory, then you can go for the Array data structure.

Two Types of Data Structures :

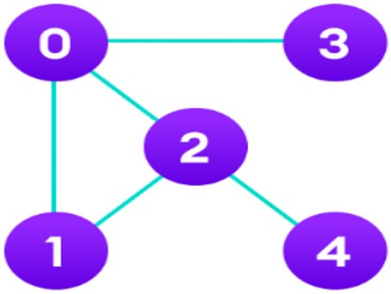
1. Linear :
    1. Array
    2. Stack
    3. Queue
    4. LinkedList

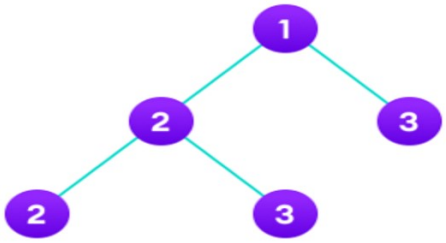
2. Non Linear :
    1. Graph
    2. Trees



LIFO (Last in First Out.)

FIFO
(First in First Out)





Basic Mathematical Concepts used in Algorithms

1. Basic Arithmetic Operations :

- Addition : a + b;
- Subtraction - a - b;
- multiplication - a * b;
- Division - a / b ;
- Modulus - a % b -> remainder.
- Exponential - a ** 2 , a ^ 2, Math.pow(a,b);

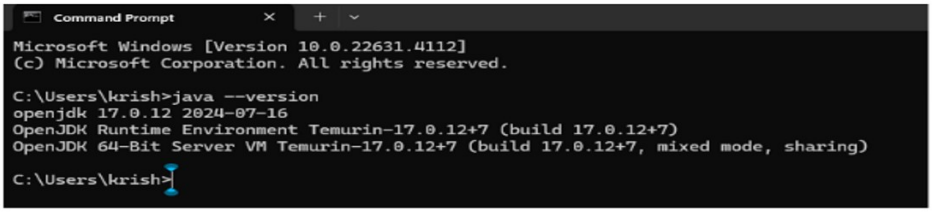
2. Comparison Operators : [Boolean T/F]
   - Equal To ("==")
   - Not Equal To ("!=")
   - Greater Than(>) , Less Than("<")
   - >= , <=.

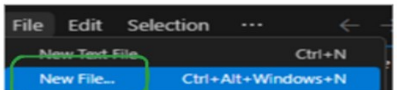
3. Logical Operator:
   - AND Logical (&&)
     - Returns True if both statement are true.
     - Eg : 5 > 3 && 7!=7 -
   - OR Logical (||)
     - Returns True if at-least one statement is True.
   - NOT(!)
     - Negate the value

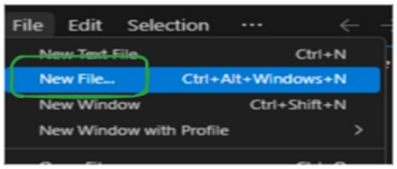
4. Built In Functions :
   Math.abs(-321) = 321
   Math.sqrt(2500) = 50
   Math.sin() , Math.cos().
   Math.log()
   Math.ceil() , Math.floor()

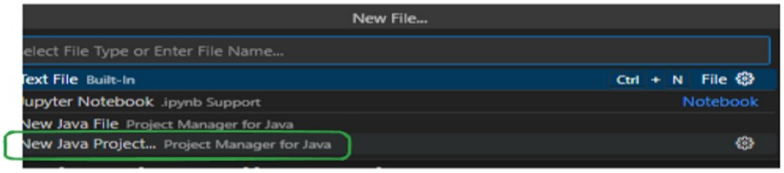
Java JDK Installation To VSCode

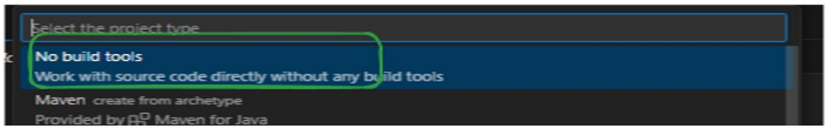
Step 1 : Install the VSCode to its default settings.
Step 2 : Search for Java Packages

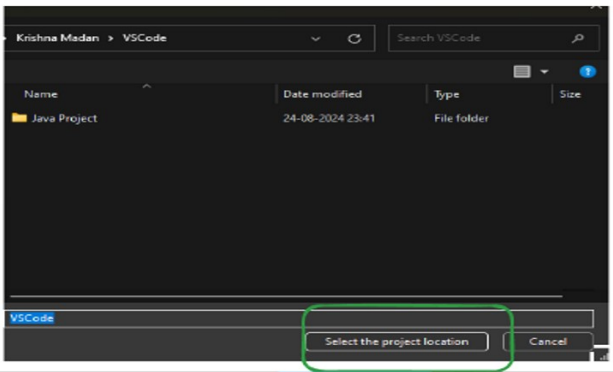


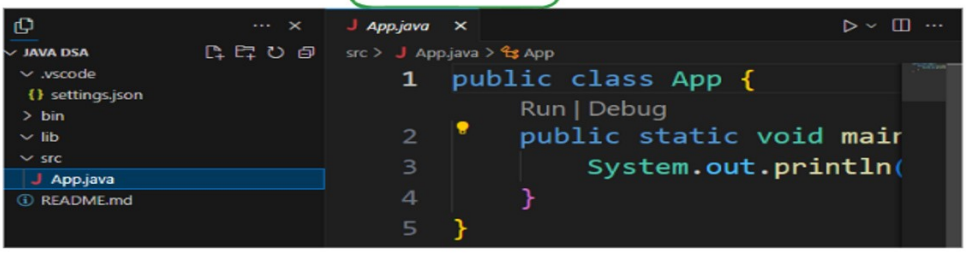


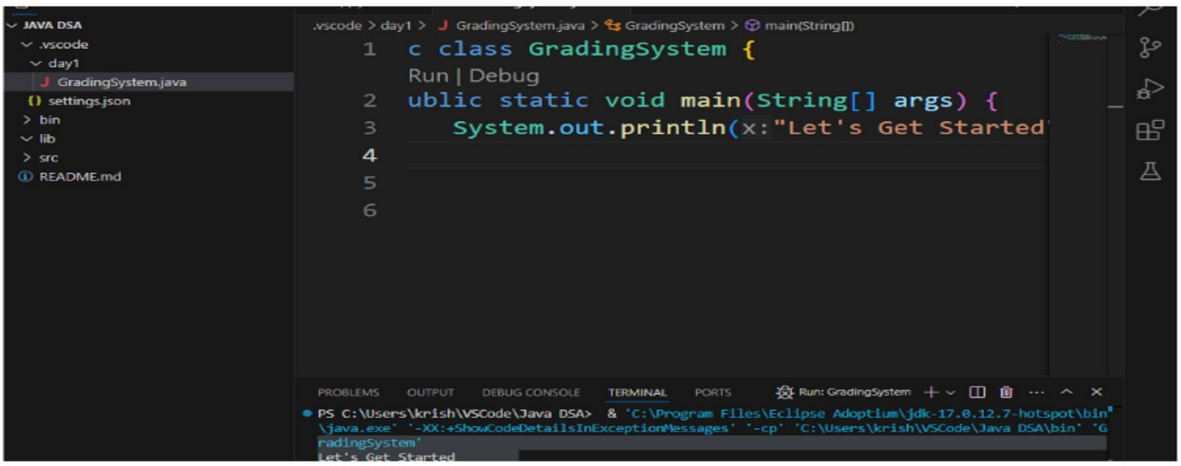












Problem Statement

Problem 1: Grading System

Write a program that takes a student's score as input and determines their grade based on the following criteria:

- Score 90 and above: Grade A
- Score between 80 and 89: Grade B
- Score between 70 and 79: Grade C

Example

Input 1:

- Score 90 and above: Grade A
- Score between 80 and 89: Grade B
- Score between 70 and 79: Grade C
- Score between 60 and 69: Grade D
- Score below 60: Grade F

Objective: Given a score, the program should print the corresponding grade.

Example

Input 1:

- Score: `85`

Output 1:

- Grade: `B`

Input 2:

- Score: `72`

Output 2:

- Grade: `C`

```
import java.util.Scanner;
public class GradingSystem {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        System.out.println("Enter the Student's Score :");
        int score = scn.nextInt();
        char grade;
        if(score >=90){
            grade = 'A';
        }else if(score >=80){
            grade = 'B';
        }else if(score >=70){
            grade = 'C';
        }else if(score >=60){
            grade = 'D';
        }else{
            grade = 'F';
        }

        System.out.println("The Student's grade is : " + grade);
    }
}
```

Problem 2: Even or Odd Number

Problem Statement:
Write a program that takes an integer as input and determines whether the number is even or odd.

Objective: Given an integer, the program should print "Even" if the number is even, and "Odd" if the number is odd.

Example

Input 1:

- Number: `4`

Output 1:

- Result: `Even`

Input 2:

- Number: `7`

Output 2:

- Result: `Odd`

```
import java.util.Scanner;
public class EvenOddChecker {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);

        System.out.println("Enter a Number:");
        int num = scn.nextInt();

        if(num % 2 ==0){
            System.out.println("Even");
        }else{
```

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

Problem 3: Positive, Negative, or Zero

Problem Statement:
Write a program that takes an integer as input and determines if it is positive, negative, or zero.

Objective: Given an integer, the program should print "Positive", "Negative", or "Zero" based on the input.

Example

Input 1:

- Number: `15`

Output 1:

- Result: `Positive`

Input 2:

- Number: `-8`

Output 2:

- Result: `Negative`

Input 3:

- Number: `0`

Output 3:

- Result: `Zero`

Attendance Code: E2E26E3F

```
import java.util.Scanner;
public class NumberSignChecker {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int num = scn.nextInt();

        if(num > 0){
            System.out.println("Positive");
        }else if(num < 0) {
            System.out.println("Negative");
        }else{
            System.out.println("Zero");
        }
    }
}
```

Problem 4: Maximum of Three Numbers

Problem Statement:
Write a program that takes three integers as input and determines the largest of the three.

Objective: Given three integers, the program should print the largest number.

Example

Input 1:

- Numbers: `5, 9, 3`

Input 1:

- Numbers: `5, 9, 3`

Output 1:

- Result: `9`

Input 2:

- Numbers: `12, 7, 12`

Output 2:

- Result: `12`

```java
import java.util.Scanner;
public class MaximumOfThree {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);

        System.out.println("Enter 3 numbers: ");

        int num1 = scn.nextInt();
        int num2 = scn.nextInt();
        int num3 = scn.nextInt();

        int max;

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

        System.out.println("The maximum number is " + max);
    }
}
```

Memory

num1= 5
num2 = 9
num3 = 3
max = 9