JABATAN KEJURUTERAAN MEKANIK
Department of Mechanical Engineering

LAPORAN PROJEK

Project Report

Sesi & Sem ***Session & Sem***	2024/2025 Semester 1
Kod & Nama Kursus ***Course Code & Name***	KIG2013 COMPUTER PROGRAMMING
Tajuk Kerja ***Title of Assignment***	ENGINEERING MODEL C++ PROGRAMMING
Nama dan No. Matriks ***Names and Matrix No.***	1. MOHAMAD FIRDAUS HADZWAN BIN MOHAMAD SYARUL (22001684) 2. MUHAMMAD LUQMAN HAKIM BIN ALI SYAHBANA (22001519) 3. MUHAMMAD ZULKHAIRIE RAHMAN BIN HATAMAL (23005150) 4. NURUL HUSNA BINTI MOHAMAD ANUAR (22001132) 5. WAN NURAISYAH KAMILIA BINTI WAN KAMIL (22002483)
Pensyarah ***Lecturer***	DR. MOHAMMED ABDO HASHEM ALI
Tarikh Hantar ***Submission Date***	20 Jan 2025

Table of Contents

1.0 Abstract

This project aims to develop a C++ program to analyse internal pipe flow characteristics, focusing on key engineering parameters such as Reynolds number, friction factor, head loss, pressure drop, and pumping work. The primary objective is to provide an efficient computational tool for solving fluid mechanics problems related to internal flows, particularly in pipes with varying dimensions, fluid properties, and flow conditions. The program employs object-oriented programming (OOP) principles, with a modular design that incorporates three files: main.cpp, internalFlow.cpp, and InternalFlow.h. The InternalFlow class encapsulates data and methods required for the analysis, using private attributes to store input parameters and computational results, while public methods execute calculations such as density determination, dynamic viscosity selection, and flow property analysis. Arrays are utilized to handle multiple cases, enabling simultaneous analysis of up to ten scenarios. The methodology involves gathering user inputs for pipe and flow characteristics, performing step-by-step computations for each case, and presenting results in a clear, structured output. The program's modular structure ensures scalability, reusability, and ease of debugging, highlighting its engineering focus on fluid dynamics and its programming emphasis on structured, reusable code. The final output includes critical insights into flow behaviour, providing engineers with a reliable tool for designing and optimizing fluid systems.

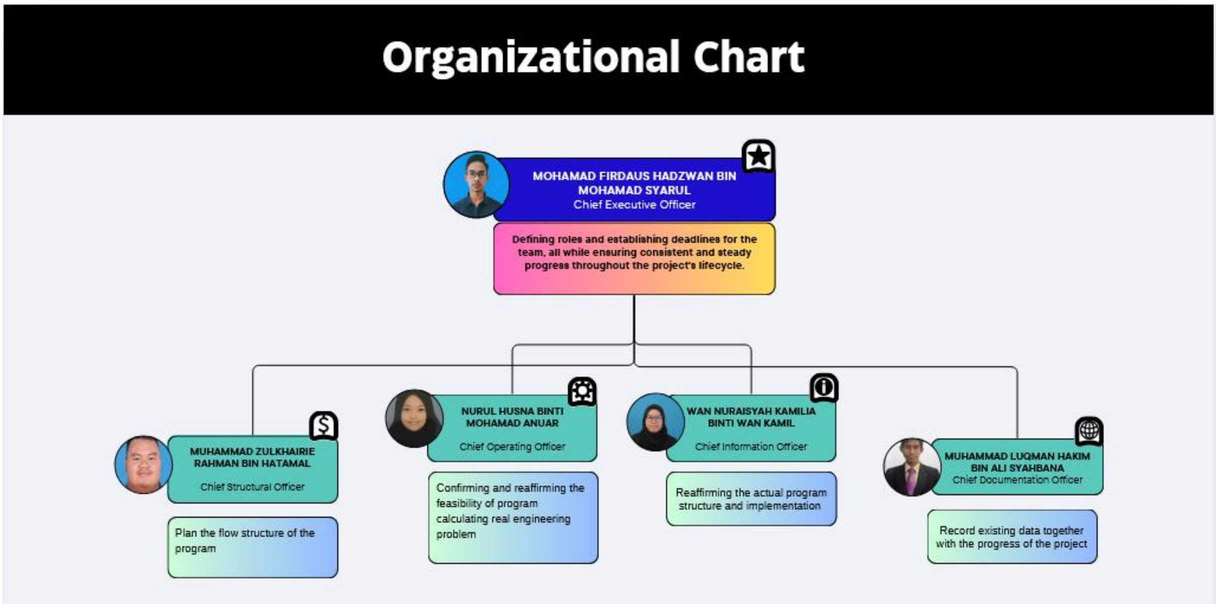
2.0 Introduction

In engineering, complicated calculations are being made in real time as to solve the problems at hand. As technology progressed, we have more access to other means of calculations other than manual calculation. One thing to take note is, with technology, we could also reduce the time for our calculations and even conduct graphical simulations.

Hence, why the use of programming and coding languages like C++ has proven useful in recent times. C++ is a powerful, high-performance programming language that is widely used for system software, application software, game development, and other areas where performance and efficiency are critical. It was created by **Bjarne Stroustrup** in the early 1980s as an extension of the C programming language, adding object-oriented programming (OOP) and other advanced features.

The purpose of this assignment is to apply C++ programming language into real life engineering problems. Our group aims to construct a programme to calculate Reynold's Number, head loss, pumping power and pressure drop using object-oriented programming in C++ language. The significance of using C++ here is to show and prove that C++ can solve engineering problems which requires complex mathematical calculations and a set of equations within a short range of time.

3.0 Team Structure



4.0 Methodology

4.1 List of Nomenclature

Vavg = average flow velocity (m/s)
D = diameter (m)
ρ = density (kg/m3)
μ = dynamic viscosity (kg/ms)
v = μ/ρ = kinematic viscosity of the fluid or viscous diffusivity or diffusivity for momentum (m2/s)
Re = Reynolds number
ρV2 avg/2 is the dynamic pressure
f = Darcy friction factor
V. = volume flow rate
m. = mass flow rate
Ac = cross-sectional area
u(r) = velocity profile
e = roughness of the pipe
hL = additional irreversible head loss in the piping system

4.2 Theoretical Background

Liquid or gas flow through pipes or ducts is commonly used in heating and cooling applications and fluid distribution networks. The fluid in such applications is usually forced to flow by a fan or pump through a flow section. We pay particular attention to friction, which is directly related to the pressure drop and head loss during flow through pipes and ducts. The pressure drop is then used to determine the pumping power requirement. A typical piping system involves pipes of different diameters connected to each other by various fittings or elbows to route the fluid, valves to control the flow rate, and pumps to pressurize the fluid. In general, flow sections of circular cross section are referred to as pipes especially when the fluid is a liquid, and flow sections of non-circular cross section as ducts especially when the fluid is a gas. Pipes with a circular cross section can withstand large pressure differences between the inside and the outside without undergoing significant distortion.

The fluid velocity in a pipe changes from zero at the surface because of the no-slip condition to a maximum at the pipe centre. In fluid flow, it is convenient to work with an average velocity Vavg which remains constant in incompressible flow when the cross-sectional area of the pipe is constant. We evaluate the fluid properties at some average temperature and treat them as constants. The convenience of working with constant properties usually more than justifies the slight loss in accuracy. Also, the friction between the fluid particles in a pipe does cause a slight rise in fluid temperature as a result of the mechanical energy being converted to sensible thermal energy. But this temperature rise due to frictional heating is usually too small to warrant any consideration in calculations and thus is disregarded. The primary consequence of friction in fluid flow is pressure drop, and thus any significant temperature change in the fluid is due to heat transfer.

A careful inspection of flow in a pipe reveals that the fluid flow is streamlined at low velocities but turns chaotic as the velocity is increased above a critical value, as shown in figure below. The flow regime in the first case is said to be laminar, characterized by smooth streamlines and highly ordered motion, and turbulent in the second case, where it is characterized by velocity fluctuations and highly disordered motion.

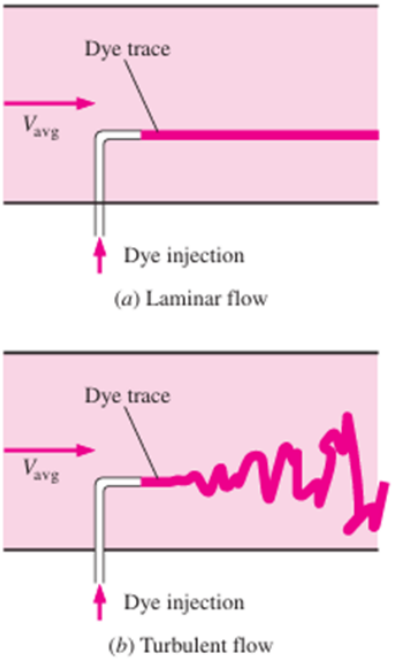


Figure shows the laminar and turbulent flow in pipes

The transition from laminar to turbulent flow does not occur suddenly rather, it occurs over some region in which the flow fluctuates between laminar and turbulent flows before it becomes fully turbulent. Most flows encountered in practice are turbulent. Laminar flow is encountered when highly viscous fluids such as oils flow in small pipes or narrow passages. We can verify the existence of these laminar, transitional, and turbulent flow regimes by injecting some dye streaks into the flow in a glass pipe, as the British engineer Osborne Reynolds (1842–1912) did over a century ago. We observe that the dye streak forms a straight and smooth line at low velocities when the flow is laminar and has bursts of fluctuations in the transitional regime, and zigzags rapidly and randomly when the flow becomes fully turbulent. These zigzags and the dispersion of the dye are indicative of the fluctuations in the main flow and the rapid mixing of fluid particles from adjacent layers. The intense mixing of the fluid in turbulent flow as a result of rapid fluctuations enhances momentum transfer between fluid particles, which increases the friction force on the surface and thus the required pumping power. The friction factor reaches a maximum when the flow becomes fully turbulent.

Reynolds Number is the transition from laminar to turbulent flow depends on the geometry, surface roughness, flow velocity, surface temperature, and type of fluid, among other things. Note that the Reynolds number is a dimensionless quantity. After exhaustive experiments in the 1880s, Osborne Reynolds discovered that the flow regime depends mainly on the ratio of inertial forces to viscous forces in the fluid. This ratio is called the Reynolds number and is expressed for internal flow in a circular pipe as follows;

$$\mathrm{Re} = \frac{\text{Inertial forces}}{\text{Viscous forces}} = \frac{V_{avg}D}{\nu} = \frac{\rho V_{avg}D}{\mu}$$

At large Reynolds numbers, the inertial forces, which are proportional to the fluid density and the square of the fluid velocity, are large relative to the viscous forces, and thus the viscous forces cannot prevent the random and rapid fluctuations of the fluid. At small or moderate Reynolds numbers, however, the viscous forces are large enough to suppress these fluctuations and to keep the fluid "in line.". Thus, the flow is turbulent in the first case and laminar in the second. Under most practical conditions, Reynolds number influenced the flow as below;

$$\begin{aligned} \text{Re} &\lesssim 2300 && \text{laminar flow} \\ 2300 \lesssim \text{Re} &\lesssim 4000 && \text{transitional flow} \\ \text{Re} &\gtrsim 4000 && \text{turbulent flow} \end{aligned}$$

A quantity of interest in the analysis of pipe flow is the pressure drop ΔP since it is directly related to the power requirements of the fan or pump to maintain flow. For laminar flow, the pressure difference is written as;

$$\Delta P = P_1 - P_2 = \frac{8\mu L V_{\text{avg}}}{R^2} = \frac{32\mu L V_{\text{avg}}}{D^2}$$

A pressure drop due to viscous effects represents an irreversible pressure loss, and it is called pressure loss ΔPL to emphasize that it is a loss. In practice, it is found convenient to express the pressure loss for all types of fully developed internal flows

$$\Delta P_L = f \frac{L}{D} \frac{\rho V_{\text{avg}}^2}{2}$$

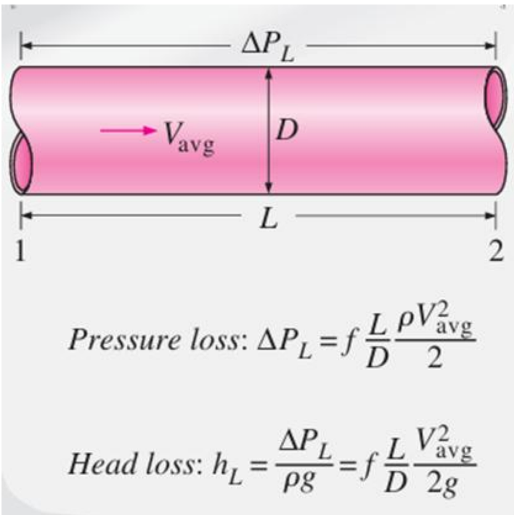


Figure shows the relation for pressure loss

f is the Darcy friction factor. It is also called the Darcy–Weisbach friction factor, named after the Frenchman Henry Darcy (1803–1858) and the German Julius Weisbach (1806–1871), the two engineers who provided the greatest contribution in its development. This equation shows that in laminar flow, the friction factor is a function of the Reynolds number only and is independent of the roughness of the pipe surface.

$$f = \frac{8\tau_w}{\rho V_{\text{avg}}^2}$$

$$f = \frac{64\mu}{\rho D V_{\text{avg}}} = \frac{64}{\text{Re}}$$

In the analysis of piping systems, pressure losses are commonly expressed in terms of the equivalent fluid column height, called the head loss, hL. The head loss hL represents the additional height that the fluid needs to be raised by a pump in order to overcome the frictional losses in the pipe. The head loss is caused by viscosity, and it is directly related to the wall shear stress.

$$h_L = \frac{\Delta P_L}{\rho g} = f \frac{L}{D} \frac{V_{\text{avg}}^2}{2g}$$

Once the pressure loss (or head loss) is known, the required pumping power to overcome the pressure loss is determined from;

$$\dot{W}_{\text{pump}, L} = \dot{V} \Delta P_L = \dot{V} \rho g h_L = \dot{m} g h_L$$

The friction factor in fully developed turbulent pipe flow depends on the Reynolds number and the relative roughness e/D, which is the ratio of the mean height of roughness of the pipe to the pipe diameter. The functional form of this dependence cannot be obtained from a theoretical analysis and all available results are obtained from painstaking experiments using artificially roughened surfaces. The friction factor was calculated from the measurements of the flow rate and the pressure drop. The experimental results obtained are presented in tabular, graphical and functional forms obtained by curve-fitting experimental data called the Moody chart. In 1939, Cyril F. Colebrook (1910–1997) combined the available data for transition and turbulent flow in smooth as well as rough pipes into the following implicit relation known as the Colebrook equation;

$$\frac{1}{\sqrt{f}} = -2.0 \log\left(\frac{\varepsilon/D}{3.7} + \frac{2.51}{\text{Re}\sqrt{f}}\right) \qquad \text{(turbulent flow)}$$

The Colebrook equation is implicit in f,and thus the determination of the friction factor requires some iteration unless an equation solver such as EES is used. An approximate explicit relation for f was given by S. E. Haaland in 1983 as;

$$\frac{1}{\sqrt{f}} \cong -1.8 \log\left[\frac{6.9}{\text{Re}} + \left(\frac{\varepsilon/D}{3.7}\right)^{1.11}\right]$$

Commercially available pipes differ from those used in the experiments in that the roughness of pipes in the market is not uniform and it is difficult to give a precise description of it. Equivalent roughness values for some commercial pipes are given below. But it should be kept in mind that these values are for new pipes and the relative roughness of pipes may increase with use because of corrosion, scale build-up and precipitation.

Equivalent roughness values for new commercial pipes*

Material	Roughness, ε ft	Roughness, ε mm
Glass, plastic	0 (smooth)	
Concrete	0.003–0.03	0.9–9
Wood stave	0.0016	0.5
Rubber, smoothed	0.000033	0.01
Copper or brass tubing	0.000005	0.0015
Cast iron	0.00085	0.26
Galvanized iron	0.0005	0.15
Wrought iron	0.00015	0.046
Stainless steel	0.000007	0.002
Commercial steel	0.00015	0.045

Figure shows the roughness of materials

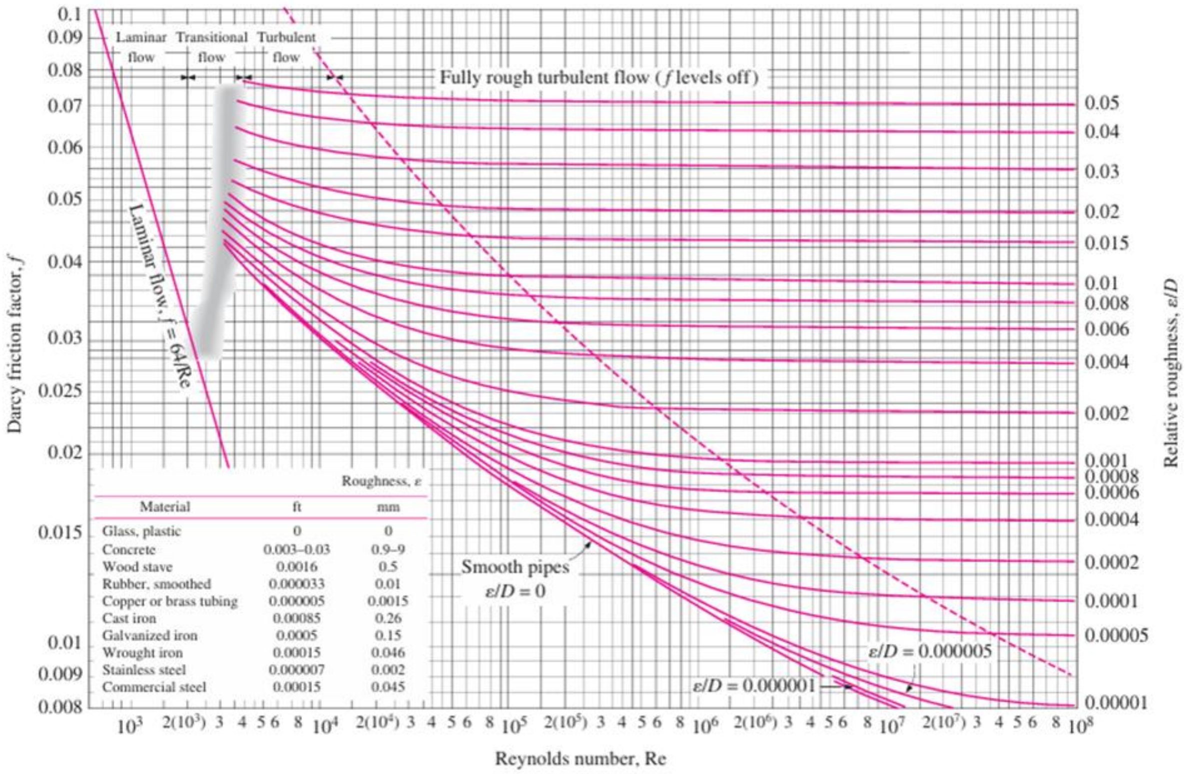


Figure shows the Moody chart

The fluid in a typical piping system passes through various fittings, valves, bends, elbows, tees, inlets, exits, enlargements and contractions in addition to the pipes. These components interrupt the smooth flow of the fluid and cause additional losses because of the flow separation and mixing they induce. In a typical system with long pipes, these losses are minor compared to the total head loss in the pipes and are called minor losses. Flow through valves and fittings is very complex, and a theoretical analysis is generally not plausible. Therefore, minor losses are determined experimentally, usually by the manufacturers of the components. Minor losses are usually expressed in terms of the loss coefficient KL, also called the resistance coefficient defined as;

$$K_L = \frac{h_L}{V^2/(2g)}$$

When the loss coefficient for a component is available, the head loss for that component is determined from;

$$h_L = K_L \frac{V^2}{2g}$$

Total head loss is then reduced to;

$$h_{L,\,\text{total}} = \left(f\frac{L}{D} + \sum K_L\right)\frac{V^2}{2g}$$

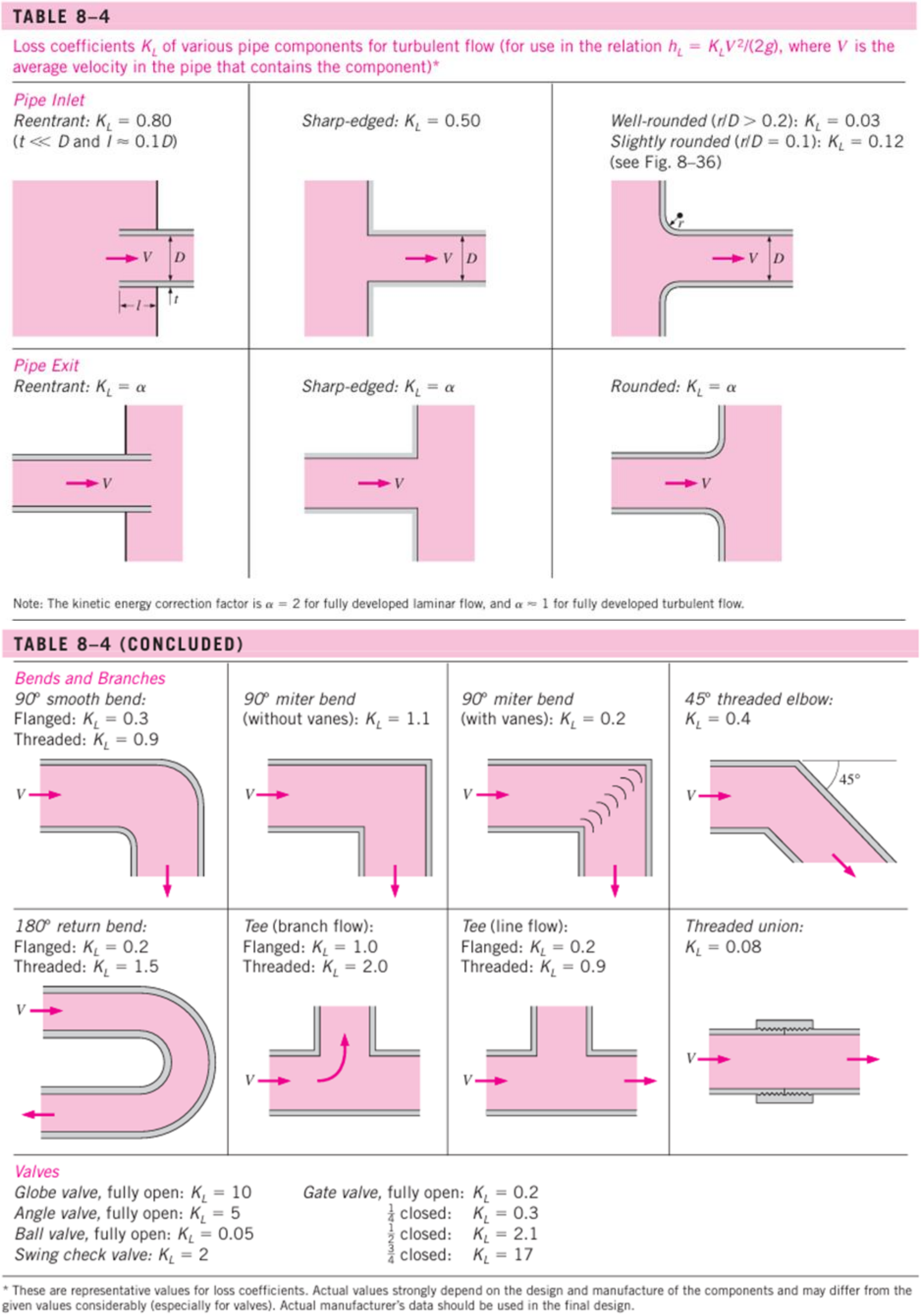


TABLE 8–4

Loss coefficients K_L of various pipe components for turbulent flow (for use in the relation $h_L = K_L V^2/(2g)$, where V is the average velocity in the pipe that contains the component)*

Pipe Inlet

Reentrant: $K_L = 0.80$ ($t \ll D$ and $l \approx 0.1D$)

Sharp-edged: $K_L = 0.50$

Well-rounded ($r/D > 0.2$): $K_L = 0.03$
Slightly rounded ($r/D = 0.1$): $K_L = 0.12$
(see Fig. 8–36)

Pipe Exit

Reentrant: $K_L = \alpha$

Sharp-edged: $K_L = \alpha$

Rounded: $K_L = \alpha$

Note: The kinetic energy correction factor is $\alpha = 2$ for fully developed laminar flow, and $\alpha \approx 1$ for fully developed turbulent flow.

TABLE 8–4 (CONCLUDED)

Bends and Branches

90° smooth bend:
Flanged: $K_L = 0.3$
Threaded: $K_L = 0.9$

90° miter bend (without vanes): $K_L = 1.1$

90° miter bend (with vanes): $K_L = 0.2$

45° threaded elbow: $K_L = 0.4$

180° return bend:
Flanged: $K_L = 0.2$
Threaded: $K_L = 1.5$

Tee (branch flow):
Flanged: $K_L = 1.0$
Threaded: $K_L = 2.0$

Tee (line flow):
Flanged: $K_L = 0.2$
Threaded: $K_L = 0.9$

Threaded union: $K_L = 0.08$

Valves

Globe valve, fully open: $K_L = 10$
Angle valve, fully open: $K_L = 5$
Ball valve, fully open: $K_L = 0.05$
Swing check valve: $K_L = 2$

Gate valve, fully open: $K_L = 0.2$
$\frac{1}{4}$ closed: $K_L = 0.3$
$\frac{1}{2}$ closed: $K_L = 2.1$
$\frac{3}{4}$ closed: $K_L = 17$

* These are representative values for loss coefficients. Actual values strongly depend on the design and manufacture of the components and may differ from the given values considerably (especially for valves). Actual manufacturer's data should be used in the final design.

Figure shows the loss coefficient for pipes component

The equations involved in this program are as follows;

1. Reynolds number

$$\text{Re} = \frac{\text{Inertial forces}}{\text{Viscous forces}} = \frac{V_{\text{avg}}D}{\nu} = \frac{\rho V_{\text{avg}}D}{\mu}$$

2. Friction factor

$$f = \frac{64\mu}{\rho D V_{\text{avg}}} = \frac{64}{\text{Re}}$$

for Reynolds number less than 2300

$$\frac{1}{\sqrt{f}} \cong -1.8 \log\left[\frac{6.9}{\text{Re}} + \left(\frac{\varepsilon/D}{3.7}\right)^{1.11}\right]$$

for Reynolds number more than 2300

3. Volume flow rate

$$\dot{V} = V_{\text{avg}}A_c = V_{\text{avg}}(\pi D^2/4)$$

4. Head loss

$$h_L = \frac{\Delta P_L}{\rho g} = f\frac{L}{D}\frac{V_{\text{avg}}^2}{2g}$$

5. Pressure drop

$$\Delta P_L = f\frac{L}{D}\frac{\rho V_{\text{avg}}^2}{2}$$

6. Pumping power

$$\dot{W}_{\text{pump},\,L} = \dot{V}\,\Delta P_L = \dot{V}\rho g h_L = \dot{m} g h_L$$

4.3 IPO Table

Input	Process	Output
Number of cases (up to 10)	Prompt the user for the number of cases, subsequently determining number of temperatures, materials, pipe diameters, lengths, and velocities that are going to be obtained.	None during input; results displayed after calculations.
Temperature for each case	Calculate water density and dynamic viscosity for each case based on temperature.	Per case: Reynolds number, head loss, pressure drop, pumping power, and volume flow rate (The values obtained are used to further calculate the mentioned values).
Material type for each case (1–10), determining pipe roughness (epsilon)	Uses in the calculation for coefficient of losses	Coefficient of losses for each case
Pipe diameter for each case	Calculate Reynolds number for each case.	Reynold Number for each case, subsequently determining type of flow in each case (Laminar or Turbulent)
Pipe length for each case	Calculate friction factor based on Reynolds number and pipe roughness ratio (epsilon/diameter).	Friction factor for each case
Average velocity for each case	Perform the main calculation: volume flow rate, head loss, pressure drop, and pumping power using fluid dynamics formulas.	Value of volume flow rate, head loss, pressure drop, and pumping power for each case
User choice of which case to display	Display the calculated results for the selected case.	Results displayed in a formatted manner based on user-selected case.

4.4 Development Tools

In this program, C++ is used as it provides capabilities for object-oriented programming (OOP) which gives a clear structure to programs and objects are essentially reusable software components to lower development costs. It helps in organizing and managing complex software projects. OOP principles such as inheritance, polymorphism, and encapsulation allow developers to create reusable and maintainable code. C++ is known for its high performance and efficiency. It allows developers to write code that can directly manipulate hardware and system resources, making it ideal for applications that require low-level programming and fine-tuned performance. C++ has a rich Standard Template Library (STL) that provides a wide range of pre-built functions, algorithms, and data structures. The STL helps developers write efficient and compact code, reducing development time and effort. It includes components like vectors, lists, maps, and sets, which are essential for various programming tasks. Hence, C++ is a versatile programming language that offers high performance, efficiency, and a rich set of features to help achieving an organised complex program.

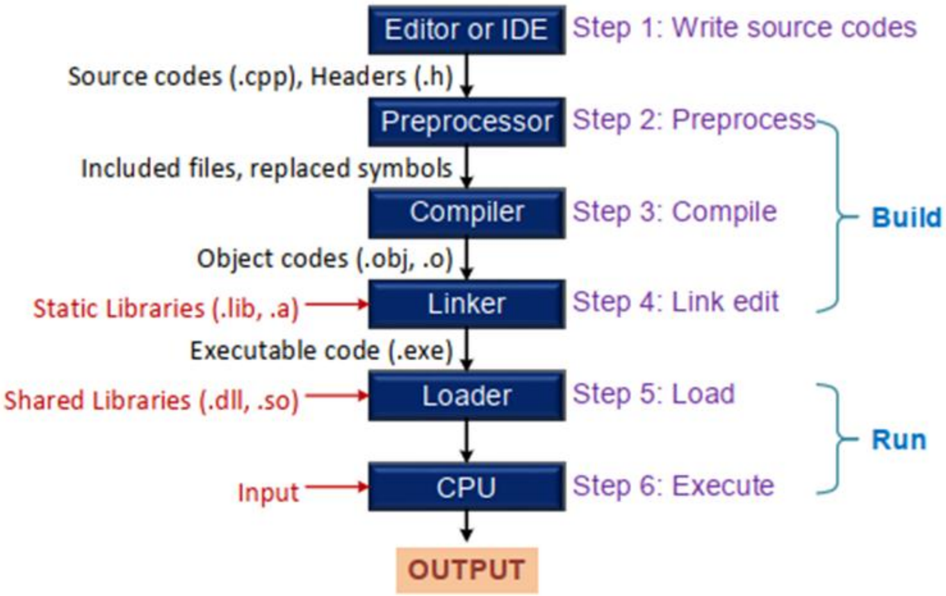
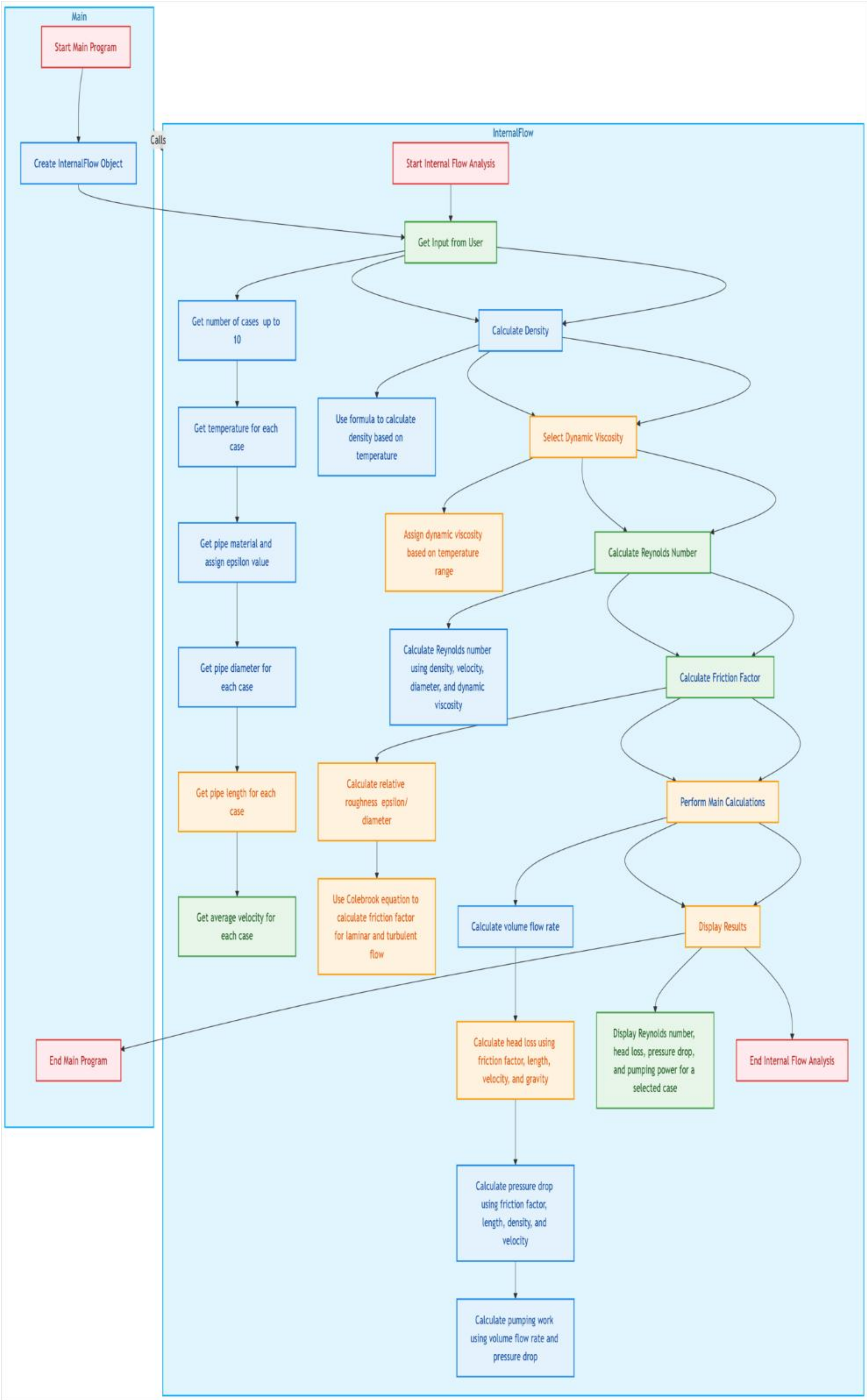


Figure shows the process of building a C++ program

5.0 Project Gantt Chart

ID	Task	Start Date	Due Date	Duration	2024/2025											
					W 9	W 9	W 10	W 10	W 11	W 11	W 12	W 12	W 13	W 13	W 14	W 14
1	Project Initiation	09/12/2024	15/12/2024	1 week												
2	First discussion	19/12/2024	19/12/2024	30 mins												
3	Prototyping	23/12/2024	27/12/2024	5 days												
4	Second discussion	28/12/2024	28/12/2024	30 mins												
5	Compiling codes	30/12/2024	03/01/2025	5 days												
6	Testing	30/12/2024	03/01/2025	5 days												
7	Report writing and documentation	28/12/2024	11/01/2025	15 days												
8	Final audit	10/01/2025	19/01/2025	10 days												
9	Record video	13/01/2025	19/01/2025	7 days												
10	Submit report, video, and pptx slides	13/01/2025	19/01/2025	7 days												

6.0 Program Flowchart



7.0 Program Analysis

7.1 Overview of the Program

The program is designed to perform detailed calculations for analysing internal pipe flow. It determines key parameters such as Reynolds number, head loss, pressure drop, and pumping power. The program utilizes a modular and object-oriented approach, leveraging a multi-file structure to organize and enhance the readability, maintainability, and scalability of the code.

7.2 Key Features

The program includes several well-implemented features:

1. Material Selection with Roughness Values:
   Arrays store the roughness values corresponding to different pipe materials (e.g., PVC, steel, cast iron), allowing quick retrieval based on user selection.
2. Dynamic Fluid Properties:
   Fluid properties such as density and viscosity are computed based on user-input temperature, ensuring accurate calculations for real-world applications.
3. Modular Design with Reusable Functions:
   Functions are designed to handle specific tasks like Reynolds number calculation, head loss determination, and pumping power estimation, ensuring clarity and simplicity.
4. Error Handling and Validation:
   Input validation mechanisms prevent users from entering unrealistic or out-of-range values, ensuring the program's reliability.

7.3 Use of Arrays

Arrays are used to store predefined data sets that are referenced multiple times in the program. For example:

- Roughness Values: An array stores roughness coefficients (ϵ\epsilonϵ) corresponding to pipe materials like PVC, steel, and cast iron. When the user selects a material, the program retrieves the associated value from the array using an index.
- Material Names: Another array contains the names of materials, enabling the program to display user-friendly options and match them with corresponding roughness values.

This approach simplifies data handling, reduces redundancy, and allows for efficient processing.

7.4 Object-Oriented Programming (OOP): The InternalFlow Class

The program is designed around the InternalFlow class, which encapsulates all the attributes and methods necessary to calculate and analyse internal pipe flow. The InternalFlow class follows the principles of object-oriented programming (OOP), ensuring data encapsulation, modularity, and reusability.

Class Structure

The InternalFlow class is defined with the following components:

1. Private Attributes:
   These store the essential data for multiple flow cases. Arrays are used to handle up to 10 flow cases simultaneously, making the program capable of processing multiple inputs in a single execution. Key attributes include:

   - Pipe Parameters:
     - diameter[10] (pipe diameter for each case).
     - length[10] (pipe length for each case).
     - epsilon[10] (pipe roughness for each case).
   - Fluid Properties:
     - temperature[10] (fluid temperature for each case).
     - density[10] (calculated fluid density).
     - dynamicViscosity[10] (calculated fluid dynamic viscosity).
   - Flow Characteristics:
     - avgVelocity[10] (average flow velocity).
     - Re[10] (Reynolds number).
     - f[10] (friction factor).
   - Calculated Outputs:
     - headLoss[10] (calculated head loss).
     - pressureDrop[10] (calculated pressure drop).
     - pumpingWork[10] (calculated pumping work).
     - volumeFlowRate[10] (calculated volumetric flow rate).
     - eD[10] (relative roughness for each case).

2. Public Methods:
   The InternalFlow class provides public methods to interact with and manipulate these attributes. Each method focuses on a specific task, ensuring modularity and clarity.

   - InternalFlow(): A default constructor that initializes the class attributes.
   - getInput(): Collects user input for pipe and flow properties for all cases.

- calculateDensity(): Computes the fluid density for each case based on the input temperature.
- dynamicViscositySelection(): Determines the dynamic viscosity of the fluid for each case based on the input temperature.
- ReynoldNumber(): Calculates the Reynolds number using the formula
- frictionFactor(): Computes the Darcy-Weisbach friction factor f using the Colebrook equation or another approximation, based on the Reynolds number and relative roughness.
- mainCalculation(): Performs the primary calculations for head loss, pressure drop, pumping work, and volumetric flow rate.
- displayOutput(int caseIndex): Displays the results for a specific flow case in a user-friendly format.
- getNumCases(): Returns the number of cases entered by the user.

Benefits of the InternalFlow Class

The InternalFlow class encapsulates all the necessary data and logic, ensuring:

1. **Scalability**: Easy addition of new calculations or parameters.
2. **Reusability**: The class can be used in other programs or expanded to handle more complex fluid flow scenarios.
3. **Modularity**: Separating tasks into methods improves code readability and simplifies debugging or testing.

7.5 Modularization with Three-File Structure

The program follows a three-file structure comprising:

1. main.cpp: The driver file containing the main() function. This file manages user input, program flow, and displays outputs to the user. It calls the functions and methods implemented in the other files.
2. InternalFlow.cpp: The implementation file where the logic for the Pipe class methods is defined. This file ensures the program's computational functionality is encapsulated and kept separate from the user interface.
3. InternalFlow.h: The header file that declares the Pipe class and its methods. It provides the blueprint for the program's structure and ensures consistent definitions across files.

7.6 Why Use a Three-File Structure?

1. Code Organization: Separating declarations (.h), definitions (.cpp), and main program logic (main.cpp) ensures better organization and readability.

2. Reusability: The InternalFlow.h header and InternalFlow.cpp implementation can be reused in other programs without modification, demonstrating high modularity.
3. Ease of Maintenance: Changes or updates to specific parts of the program (e.g., modifying a calculation formula) can be done without affecting unrelated code.
4. Compilation Efficiency: The structure allows incremental compilation, where only modified files need to be recompiled, saving time during development.

7.7 How Does It Work?

1. The InternalFlow.h file is included in main.cpp via a #include directive, giving main.cpp access to the Pipe class and its methods.
2. The InternalFlow.cpp file provides the implementation of the methods declared in InternalFlow.h. During compilation, these files are linked together to form the final executable.

7.8 Functions Used in the Program

The program relies on the following functions to perform its tasks, with a clear division of responsibilities:

1. Constructor
   - InternalFlow()
     - Initializes the attributes of the InternalFlow class, ensuring all arrays and variables are ready for input and computation.
2. Input Handling
   - getInput()
     - Gathers user inputs for pipe dimensions, flow velocity, fluid temperature, and material type for up to 10 cases.
     - Ensures input validation to prevent errors during computation.
3. Fluid Property Calculations
   - calculateDensity()
     - Calculates the fluid density for each case based on the input temperature. This ensures temperature-dependent variations in density are accounted for.
   - dynamicViscositySelection()
     - Determines the dynamic viscosity for each case based on temperature.
4. Flow and Friction Calculations
   - ReynoldNumber()
     - Computes the Reynolds number for each case to classify the flow as laminar or turbulent.

- frictionFactor()
  - Uses the Reynolds number and relative roughness to compute the friction factor, which is crucial for head loss calculations.

5. Main Calculations
   - mainCalculation()
     - Executes the core calculations, including:
     - Head Loss: Using the Darcy-Weisbach equation
     - Pressure Drop: Converts head loss to pressure drop using fluid density and gravity.
     - Pumping Work: Calculates the power required to overcome pressure drop.
     - Volume Flow Rate: Determines the volumetric flow rate based on velocity and pipe diameter.
6. Output Display
   - displayOutput(int caseIndex)
     - Displays the results for a specific case, including all calculated parameters such as Reynolds number, friction factor, head loss, pressure drop, and pumping work, in a clear and organized format.
7. Utility Function
   - getNumCases()
     - Returns the number of cases entered by the user, ensuring the program processes only valid cases.

7.9 Three-File Structure Integration

1. main.cpp:
   - Acts as the driver of the program.
   - Initializes an InternalFlow object.
   - Calls methods such as getInput(), mainCalculation(), and displayOutput() to manage program flow and present results.
2. InternalFlow.cpp:
   - Implements the methods of the InternalFlow class.
   - Handles the computational logic, ensuring separation from user interface concerns.
3. InternalFlow.h:
   - Declares the InternalFlow class, its attributes, and methods.

- Serves as the blueprint for the program, ensuring consistent definitions across files.

This structure ensures a clean separation of concerns, simplifies debugging, and allows the computational logic (internalFlow.cpp) to be reused in other projects.

7.10 Concepts Applied in Development

The program integrates multiple programming concepts:

1. **Arrays**: Used for efficient data storage and retrieval.
2. **Control Structures**: Implemented through loops (e.g., for loops for iterative calculations) and conditional statements (if-else or switch-case) to handle decision-making processes.
3. **Functions**: Modular design achieved by splitting tasks into reusable and testable functions.
4. **OOP Principles**: The Pipe class encapsulates data and behaviour, providing a structured and extensible foundation for the program.
5. **Three-File Structure**: Separates declarations, implementations, and user interactions to improve code organization and scalability.

7.11 Strengths of the Program

1. **Scalability**: The modular design allows new features or materials to be easily integrated.
2. **Accuracy**: Incorporates temperature-dependent fluid properties and validated calculations.
3. **Usability**: The clear input-output interface and error handling mechanisms enhance the user experience.
4. **Flexibility**: The three-file structure and OOP principles make the program adaptable for more complex fluid mechanics problems in the future.

8.0 Results

8.1 Run the Program

The code is run for 10 cases as defined by user. Each parameter of input for the 10 cases are variables that is chosen randomly with 10 different values. Consequently, the parameters of output for the 10 cases are displayed with 10 different values. After the program is run, the overall result displayed is as below.

*** Engineering Model: Internal Water Flow in Pipe Analysis ***

Enter the number of cases (up to 10): 10

Enter the temperature (degree Celsius) for each case:

Case 1 Temperature: 10

Case 2 Temperature: 25

Case 3 Temperature: 40

Case 4 Temperature: 55

Case 5 Temperature: 70

Case 6 Temperature: 85

Case 7 Temperature: 100

Case 8 Temperature: 115

Case 9 Temperature: 8

Case 10 Temperature: 5

Select epsilon value for pipe material from the following options:

Materials Epsilon (mm)

1) Glass, plastic 0

2) Concrete 0

3) Wood stave 0.5

4) Rubber, smoothed 0.01

5) Copper/ Brass tubing 0.0015

6) Cast iron 0.26

7) Galvanized iron 0.15

8) Wrought iron 0.046

9) Stainless steel 0.002

10) Commercial steel 0.045

Case 1 Material (Enter 1-10): 2

Case 2 Material (Enter 1-10): 3

Case 3 Material (Enter 1-10): 4

Case 4 Material (Enter 1-10): 5

Case 5 Material (Enter 1-10): 6

Case 6 Material (Enter 1-10): 7

Case 7 Material (Enter 1-10): 8

Case 8 Material (Enter 1-10): 9

Case 9 Material (Enter 1-10): 10

Case 10 Material (Enter 1-10): 1

Enter the pipe diameter (m) for each case:

Case 1 Diameter: 0.0012

Case 2 Diameter: 0.0127

Case 3 Diameter: 0.01905

Case 4 Diameter: 0.0254

Case 5 Diameter: 0.03175

Case 6 Diameter: 0.0508

Case 7 Diameter: 0.06016

Case 8 Diameter: 0.0762

Case 9 Diameter: 0.004

Case 10 Diameter: 0.0036

Enter the pipe length (m) for each case:

Case 1 Length: 15

Case 2 Length: 10

Case 3 Length: 20

Case 4 Length: 25

Case 5 Length: 13

Case 6 Length: 18

Case 7 Length: 21

Case 8 Length: 12

Case 9 Length: 16

Case 10 Length: 30

Enter the average velocity (m/s) for each case:

Case 1 Velocity: 0.9

Case 2 Velocity: 0.8

Case 3 Velocity: 1

Case 4 Velocity: 1.2

Case 5 Velocity: 1.8

Case 6 Velocity: 0.047

Case 7 Velocity: 0.038

Case 8 Velocity: 0.022

Case 9 Velocity: 0.51

Case 10 Velocity: 0.67

Enter the case number to view (1 to 10, or 0 to exit): 1

Case 1 Results:

Reynold number: 825.9444

Head Loss: 39.9876 m

Pressure Drop: 392100.0000 Pa

Pumping Power: 0.3991 W

Enter the case number to view (1 to 10, or 0 to exit): 2

Case 2 Results:

Reynold number: 11365.6245

Head Loss: 1.7133 m

Pressure Drop: 16752.6521 Pa

Pumping Power: 1.6977 W

Enter the case number to view (1 to 10, or 0 to exit): 3

Case 3 Results:

Reynold number: 28933.5220

Head Loss: 1.3219 m

Pressure Drop: 12861.6967 Pa

Pumping Power: 3.6659 W

Enter the case number to view (1 to 10, or 0 to exit): 4

Case 4 Results:

Reynold number: 59580.5056

Head Loss: 1.4518 m

Pressure Drop: 14031.2531 Pa

Pumping Power: 8.5317 W

Enter the case number to view (1 to 10, or 0 to exit): 5

Case 5 Results:

Reynold number: 138235.8381

Head Loss: 2.4338 m

Pressure Drop: 23331.4245 Pa

Pumping Power: 33.2500 W

Enter the case number to view (1 to 10, or 0 to exit): 6

Case 6 Results:

Reynold number: 6940.6170

Head Loss: 0.0015 m

Pressure Drop: 14.2250 Pa

Pumping Power: 0.0014 W

Enter the case number to view (1 to 10, or 0 to exit): 7

Case 7 Results:

Reynold number: 7763.9356

Head Loss: 0.0009 m

Pressure Drop: 8.2050 Pa

Pumping Power: 0.0009 W

Enter the case number to view (1 to 10, or 0 to exit): 8

Case 8 Results:

Reynold number: 6838.5867

Head Loss: 0.0001 m

Pressure Drop: 1.2408 Pa

Pumping Power: 0.0001 W

Enter the case number to view (1 to 10, or 0 to exit): 9

Case 9 Results:

Reynold number: 1560.3915

Head Loss: 2.1749 m

Pressure Drop: 21330.24 Pa

Pumping Power: 0.1367 W

Enter the case number to view (1 to 10, or 0 to exit): 10

Case 10 Results:

Reynold number: 1587.6975

Head Loss: 7.6857 m

Pressure Drop: 75387.4074 Pa

Pumping Power: 0.5141 W

Enter the case number to view (1 to 10, or 0 to exit): 0

Process returned 0 (0x0)   execution time : 380.463 s

Press any key to continue.

8.2 Input

Number of cases (up to 10): 10

The values of input parameters for the 10 cases are defined as shown in Table 2.

Table 2: Values of Input of Program for the 10 Cases

Case	Temperature (°C)	Pipe Material (1-10)	Diameter (m)	Pipe Length (m)	Average Velocity (m/s)
1	10	Concrete (2)	0.0012	15	0.9
2	25	Wood stave (3)	0.0127	10	0.8
3	40	Rubber, smoothed (4)	0.01905	20	1
4	55	Copper/ Brass tubing (5)	0.0254	25	1.2
5	70	Cast iron (6)	0.03175	13	1.8
6	85	Galvanized iron (7)	0.0508	18	0.047
7	100	Wrought iron (8)	0.06016	21	0.038
8	115	Stainless steel (9)	0.0762	12	0.022
9	8	Commercial steel (10)	0.04	16	0.1
10	5	Glass, plastic (1)	0.0036	30	0.67

8.3 Output

The values of output parameters for the 10 cases are displayed as shown in Table 3.

Table 3: Values of Output of Program for the 10 Cases

Case	Reynold Number	Head Loss (m)	Pressure Drop (Pa)	Pumping Power (W)
1	825.9444	39.9876	392100	0.3991
2	11365.6245	1.7133	16757.0412	1.6984
3	28933.5220	1.3219	12861.6967	3.6659
4	59580.5056	1.4518	14031.2531	8.5317
5	138235.8381	2.4338	23331.4245	33.2500
6	6940.617	0.0015	14.2250	0.0014
7	7763.9356	0.0009	8.2050	0.0009
8	6838.5867	0.0001	1.2408	0.0001
9	1560.3915	2.1749	21330.24	0.1367
10	1587.6975	7.6857	75387.4074	0.5141

8.4 Manual Calculation in Determining Output for Case 1 to Case 10

Case 1:

$$Re = \frac{(999.7)(0.9)(0.0012)0.001307999.70.90.0012}{0.001307} = 826.0719$$

Reynolds number < 2300

$$Friction\ factor = \frac{64}{826.0719} \ = 0.077475094$$

$$Volume\ flow\ rate = (0.9)(\tfrac{\pi(0.0012^2)}{4}) = 1.01801 \times 10^{-6}$$

$$Head\ loss = \ 0.077475094\left(\frac{15}{0.0012}\right)\left(\frac{0.9^2}{9.81}\right) = \ 39.9814\ m$$

$$Pressure\ drop = 0.077475094(\tfrac{15}{0.0012})(\tfrac{(999.7)(0.9^2)}{2}) = 392100\ \ Pa$$

$$Pumping\ power\ = (1.01801 \times 10 - 6)(392100) \ = 0.3992\ W$$

Case 2:

$$Reynolds\ number\ = \frac{(997)(0.8)(0.0127)}{0.000891} = 11368.7093$$

$$Reynolds\ number > 2300$$

Using

$$\frac{1}{\sqrt{f}} = -1.8\log\left(\left(\frac{6.9}{11368.7093}\right) + \left(\frac{\frac{0.5}{0.0127}}{3.7}\right)^{1.11}\right)$$

Friction factor = 0.066704621

$$Volume\ flow\ rate\ = (0.9)\left(\frac{\pi(0.0127^2)}{4}\right) = 0.000101355\ m3/s$$

$$Head\ loss\ = 0.066704621\left(\frac{10}{0.0127}\right)\left(\frac{0.8^2}{2(9.81)}\right) = 1.7133\ m$$

$$Pressure\ drop\ = 0.066704621\left(\frac{10}{0.0127}\right)\left(\frac{(997)(0.8^2)}{2(9.81)}\right) = 16757.0412\ Pa$$

$$Pumping\ power\ = (0.000101355)(16757.0412) = 1.6984\ W$$

Case 3:

$$Reynolds\ number\ = \frac{(992.1)(1)(0.01905)}{0.000653} = 2894.5804$$

$$Reynolds\ number > 2300$$

Using

$$\frac{1}{\sqrt{f}} = -1.8\log\left(\left(\frac{6.9}{2894.5804}\right) + \left(\frac{\frac{0.01}{0.01905}}{3.7}\right)^{1.11}\right)$$

Friction factor = 0.024702814

$$Volume\ flow\ rate\ = (1)\left(\frac{\pi(0.01905^2)}{4}\right) = 0.00026506\ m3/s$$

$$Head\ loss\ = 0.024702814\left(\frac{20}{0.01905}\right)\left(\frac{1^2}{2(9.81)}\right) = 1.3218\ m$$

$$Pressure\ drop\ = 0.024702814\left(\frac{20}{0.01905}\right)\left(\frac{(992.1)(1^2)}{2}\right) = 12864.9143\ Pa$$

$$Pumping\ power\ = (0.00028506)(12864.9143) = 3.6673\ W$$

Case 4:

$$Reynolds\ number\ = \frac{(985.2)(1.2)(0.0254)}{0.000504} = 59581.1429$$

$$Reynolds\ number > 2300$$

Using

$$\frac{1}{\sqrt{f}} = -1.8 \log\left(\left(\frac{6.9}{59581.1429}\right) + \left(\frac{\frac{0.0015}{0.0254}}{3.7}\right)^{1.11}\right)$$

Friction factor = 0.020097263

$$Volume\ flow\ rate\ = (1.2)\left(\frac{\pi(0.0254^2)}{4}\right) = 0.000608128\,\frac{m^3}{s}$$

$$Head\ loss\ = 0.020097263\left(\frac{25}{0.0254}\right)\left(\frac{1.2^2}{2(9.81)}\right) = 1.4518\,m$$

$$Pressure\ drop\ = 0.020097263\left(\frac{25}{0.0254}\right)\left(\frac{(985.2)(1.2^2)}{2}\right) = 14031.3710\,Pa$$

$$Pumping\ power\ = (0.000608128)(14031.3710) = 8.5329\,W$$

Case 5:

$$Reynolds\ number\ = \frac{(977.5)(1.8)(0.03175)}{0.000404} = 138277.5371$$

$$Reynolds\ number > 2300$$

Using

$$\frac{1}{\sqrt{f}} = -1.8 \log\left(\left(\frac{6.9}{138277.5371}\right) + \left(\frac{\frac{0.26}{0.03175}}{3.7}\right)^{1.11}\right)$$

Friction factor = 0.035994752

$$Volume\ flow\ rate\ = (1.8)\left(\frac{\pi(0.03175^2)}{4}\right) = 0.0014253\,\frac{m^3}{s}$$

$$Head\ loss\ = 0.035994752\left(\frac{13}{0.03175}\right)\left(\frac{1.8^2}{2(9.81)}\right) = 2.4338\,m$$

$$Pressure\ drop\ = 0.035994752\left(\frac{13}{0.03175}\right)\left(\frac{(977.5)(0.9^2)}{2}\right) = 23338.3737\,Pa$$

$$Pumping\ power\ = (0.0014253)(23338.3737) = 33.2642\,W$$

Case 6:

$$Reynolds\ number\ = \frac{(968.1)(0.047)(0.0508)}{0.0003333} = \ 6941.2479$$

$$Reynolds\ number > 2300$$

Using

$$\frac{1}{\sqrt{f}} = -1.8\ \log\ (\ (\frac{6.9}{6941.2479}) + \left(\frac{\frac{0.15}{0.0508}}{3.7}\right))^{1.11}$$

Friction factor = 0.037547997

$$Volume\ flow\ rate\ = (0.047)\left(\frac{\pi(0.0508^2)}{4}\right) = \ 0.00009526\ \ m^3/s$$

$$Head\ loss\ \ = \ 0.037547997\left(\frac{18}{0.0508}\right)\left(\frac{0.047^2}{2(9.81)}\right) = \ 0.0015\ m$$

$$Pressure\ drop\ \ = \ 0.037547997\left(\frac{18}{0.0508}\right)\left(\frac{(968.1)(0.047^2)}{2}\right) = 14.2260\ Pa$$

$$Pumping\ power\ \ = \ (0.00009526)(14.2260)\ \ = \ 0.0014\ W$$

Case 7:

$$Reynolds\ number\ = \frac{(957.9)(0.038)(0.06016)}{0.000282} = \ 7765.376$$

$$Reynolds\ number > 2300$$

Using

$$\frac{1}{\sqrt{f}} = -1.8\ \log\ (\ (\frac{6.9}{7765.376}) + \left(\frac{\frac{0.046}{0.06016}}{3.7}\right))^{1.11}$$

Friction factor = 0.033991382

$$Volume\ flow\ rate\ = (0.038)\left(\frac{\pi(0.06016^2)}{4}\right)\ = \ 0.000108\ \ m^3/s$$

$$Head\ loss\ \ = \ 0.033991382\left(\frac{21}{0.06016}\right)\left(\frac{0.038^2}{2(9.81)}\right) = \ 0.0009\ m$$

$$Pressure\ drop\ \ = \ 0.033991382\left(\frac{21}{0.06016}\right)\frac{(957.9)(0.038^2)}{2} = 8.2061\ Pa$$

$$Pumping\ power\ \ = \ (0.000108)(8.2061)\ \ = \ 0.0009\ W$$

Case 8:

$$Reynolds\ number\ = \frac{(943.4)(0.022)(0.0762)}{0.000232} = 6816.8783$$

$$Reynolds\ number > 2300$$

Using

$$\frac{1}{\sqrt{f}} = -1.8 \log\left(\left(\frac{6.9}{6816.8783}\right) + \left(\frac{\frac{0.002}{0.0762}}{3.7}\right)^{1.11}\right)$$

Friction factor = 0.034433181

$$Volume\ flow\ rate\ = (0.022)\left(\frac{\pi(0.0762^2)}{4}\right) = 0.0001\ m^3/s$$

$$Head\ loss\ = 0.034433181\left(\frac{12}{0.0762}\right)\left(\frac{0.022^2}{2(9.81)}\right) = 0.0001\ m$$

$$Pressure\ drop\ = 0.034433181\left(\frac{12}{0.0762}\right)\left(\frac{(943.4)(0.022^2)}{2}\right) = 1.2380\ Pa$$

$$Pumping\ power\ = (0.0001)(1.2380) = 0.0001\ W$$

Case 9:

$$Reynolds\ number\ = \frac{(999.7)(0.51)(0.004)}{0.001307} = 1560.3581$$

$$Reynolds\ number < 2300$$

$$Friction\ factor\ = \frac{64}{1560.3581} = 0.041016226$$

$$Volume\ flow\ rate\ = (0.51)\left(\frac{\pi(0.004^2)}{4}\right) = 0.000006409\ m^3/s$$

$$Head\ loss\ = 0.041016226\left(\frac{16}{0.004}\right)\left(\frac{0.51^2}{2(9.81)}\right) = 2.1750\ m$$

$$Pressure\ drop\ = 0.041016226\left(\frac{16}{0.004}\right)\left(\frac{(999.7)(0.51^2)}{2}\right) = 21330.24\ Pa$$

$$Pumping\ power\ = (0.000006409)(21330.4) = 0.1367\ W$$

Case 10:

$$Reynolds\ number\ = \frac{(999.9)(0.67)(0.0036)}{0.001519} = 1587.7280$$

$$Reynolds\ number < 2300$$

$$Friction\ factor\ = \frac{64}{1587.7280} = 0.040309172$$

$$Volume\ flow\ rate\ = (0.67)\left(\frac{\pi(0.0036^2)}{4}\right) = 0.00000682\,\frac{m^3}{s}$$

$$Head\ loss\ = 0.040309172\left(\frac{30}{0.0036}\right)\left(\frac{0.67^2}{2(9.81)}\right) = 7.6855\ m$$

$$Pressure\ drop\ = 0.040309172\left(\frac{30}{0.0036}\right)\left(\frac{(999.9)(0.67^2)}{2}\right) = 75387.4074\ Pa$$

$$Pumping\ power\ = (0.00000682)(75387.4074) = 0.5141\ W$$

8.5 Table of Comparison between The Program and The Manual Calculation

The values of output parameters obtained from the program and from manual calculation are shown in Table 4 to Table 13 for Case 1 to Case 10 respectively. The two values are compared by calculating their percent difference as shown in the Percentage Difference column of each table. The percentage difference indicated how much the two values differ.

Table 4: Comparison between The Program and The Manual Calculation for Case 1

Output	Program Output	Manual Calculation	Percentage Difference (%)
Re	825.9444	826.0719	0.02
Head Loss (m)	39.9876	39.9814	0.02
Pressure Drop (Pa)	392100	392100	0.00
Pumping Power (W)	0.3991	0.3992	0.02

Table 5: Comparison between The Program and The Manual Calculation for Case 2

Output	Program Output	Manual Calculation	Percentage Difference (%)
Re	11365.62	11368.7093	0.03
Head Loss (m)	1.7133	1.7133	0.00
Pressure Drop (Pa)	16752.6521	16757.0412	0.03
Pumping Power (W)	1.6977	1.6984	0.04

Table 6: Comparison between The Program and The Manual Calculation for Case 3

Output	Program Output	Manual Calculation	Percentage Difference (%)
Re	28933.5220	28942.5804	0.03
Head Loss (m)	1.3219	1.3219	0.00
Pressure Drop (Pa)	12861.6967	12864.9143	0.03
Pumping Power (W)	3.6659	3.6673	0.04

Table 7: Comparison between The Program and The Manual Calculation for Case 4

Output	Program Output	Manual Calculation	Percentage Difference (%)
Re	59580.5056	59581.1429	0.00
Head Loss (m)	1.4518	1.4518	0.00
Pressure Drop (Pa)	14031.2531	14031.3710	0.00
Pumping Power (W)	8.5317	8.5329	0.01

Table 8: Comparison between The Program and The Manual Calculation for Case 5

Output	Program Output	Manual Calculation	Percentage Difference (%)
Re	138235.8381	138277.5371	0.03
Head Loss (m)	2.4338	2.4338	0.00
Pressure Drop (Pa)	23331.4245	23338.3737	0.03
Pumping Power (W)	33.2500	33.2642	0.04

Table 9: Comparison between The Program and The Manual Calculation for Case 6

Output	Program Output	Manual Calculation	Percentage Difference (%)
Re	6940.617	6941.2479	0.01
Head Loss (m)	0.0015	0.0015	0.00
Pressure Drop (Pa)	14.2250	14.2260	0.01
Pumping Power (W)	0.0014	0.0014	0.00

Table 10: Comparison between The Program and The Manual Calculation for Case 7

Output	Program Output	Manual Calculation	Percentage Difference (%)
Re	7763.9356	7765.3760	0.02
Head Loss (m)	0.0009	0.0009	0.00
Pressure Drop (Pa)	8.2050	8.2061	0.01
Pumping Power (W)	0.0009	0.0009	0.00

Table 11: Comparison between The Program and The Manual Calculation for Case 8

Output	Program Output	Manual Calculation	Percentage Difference (%)
Re	6838.5867	6816.8783	0.32
Head Loss (m)	0.0001	0.0001	0.00
Pressure Drop (Pa)	1.2408	1.2380	0.23
Pumping Power (W)	0.0001	0.0001	0.00

Table 12: Comparison between The Program and The Manual Calculation for Case 9

Output	Program Output	Manual Calculation	Percentage Difference (%)
Re	1560.3915	1560.3581	0.00
Head Loss (m)	2.1749	2.1750	0.00
Pressure Drop (Pa)	21330.24	21330.2400	0.00
Pumping Power (W)	0.1367	0.1367	0.00

Table 13: Comparison between The Program and The Manual Calculation for Case 10

Output	Program Output	Manual Calculation	Percentage Difference (%)
Re	1587.6975	1587.7280	0.00
Head Loss (m)	7.6857	7.6855	0.00
Pressure Drop (Pa)	75387.4074	75387.4074	0.00
Pumping Power (W)	0.5141	0.5142	0.02

9.0 Discussion

The developed program for analysing internal fluid flow in a pipe successfully calculates key parameters such as the Reynolds number, head loss, pressure drop, and pumping power required to overcome the head loss. The comparison of program-generated results with manual calculations for a specific case demonstrates a high level of accuracy, with errors remaining minimal across all parameters. For instance, the program's calculated Reynolds number deviated by only 0.0164% from the manual calculation, while the head loss showed an error of 0.0155%. The pressure drop exhibited no error, and the pumping power calculation displayed an error of 0.025%. These results highlight the robustness and reliability of the program in handling steady, incompressible, and fully developed water flow in a horizontal pipe, as per the defined assumptions.

The program's design effectively considers essential input parameters such as water temperature, pipe material, diameter, length, and average velocity, providing a flexible and user-friendly approach for multiple cases. The inclusion of a material selection feature, which assigns roughness values based on pipe material, further enhances the program's applicability across diverse scenarios. However, the simplifications, such as the assumption of negligible minor losses and the exclusion of pumps or turbines, limit the program's utility in more complex systems.

One significant strength of the program lies in its integration of water properties as functions of temperature. By employing precise correlations for water density and dynamic viscosity across the temperature range of 0°C to 150°C, the program ensures accurate calculation of flow characteristics. This feature makes the program versatile for applications in environments with varying thermal conditions.

The negligible error percentages observed in the validation process can be attributed to potential rounding differences between the program's calculations and manual computations. This suggests that the program's numerical algorithms are consistent and precise. Nonetheless, expanding the validation process to include more cases and comparing the results against experimental or industry-standard data could further reinforce the program's credibility.

Despite its strengths, the program's limitations should be addressed in future iterations. Currently, the program assumes a fully developed flow and neglects minor losses, which may not hold true in practical engineering systems with complex geometries or flow transitions. Including features to account for these factors, such as entrance effects or fittings, would enhance its real-world applicability. Furthermore, extending the program to accommodate non-horizontal pipes or multi-phase flows would broaden its functionality for more advanced engineering problems.

In conclusion, the program provides an efficient and accurate tool for analysing internal flow in pipes under the specified assumptions. The minimal discrepancies between the program's outputs and manual solutions underscore its reliability, while the integration of water property correlations enhances its applicability. However, addressing the outlined limitations and conducting more extensive validation would strengthen its utility in broader engineering contexts.

10.0 Conclusion

This project successfully developed a modular C++ program to analyse the internal flow of fluids in pipes, addressing key engineering challenges such as calculating Reynolds number, friction factor, head loss, pressure drop, pumping work, and other essential parameters. By incorporating object-oriented programming (OOP) principles, the program was structured using three files: main.cpp, InternalFlow.cpp, and InternalFlow.h, enabling modularization, code reusability, and ease of maintenance. Arrays were used effectively to handle multiple cases simultaneously, ensuring the program's efficiency in analysing various scenarios.

The program automates complex iterative calculations, streamlining the process of internal flow analysis and reducing the likelihood of human errors. Its user-friendly design allows engineers to input relevant data, compute essential flow parameters, and obtain detailed outputs for each case. The approach taken in this project demonstrates the potential of programming to solve practical engineering problems by providing a reliable, accurate, and cost-effective alternative to manual calculations and expensive commercial software.

Overall, this project highlights the integration of engineering principles and computational tools to simplify fluid dynamics analysis, making it an invaluable resource for students, researchers, and professionals in the field of fluid mechanics and engineering design.

10.0 References

1. Cengel, Y. A. & Cimbala, J. M. (2006). Fluid Mechanics Fundamentals and Applications. The McGraw-Hill Companies.
2. Jones, F., & Harris, G. (1992). ITS-90 density of water formulation for volumetric standards calibration. Journal of Research of the National Institute of Standards and Technology, 97(3), 335. https://doi.org/10.6028/jres.097.013
3. Xiao, L. (2023, February 1). What is C++ used for? Codecademy Blog. https://www.codecademy.com/resources/blog/what-is-c-plus-plus-used-for/
4. W3Schools.com. (n.d.). https://www.w3schools.com/cpp/cpp_intro.asp
5. GeeksforGeeks. (2024, December 11). Top 10 reasons to learn C++ in 2025. GeeksforGeeks. https://www.geeksforgeeks.org/top-10-reasons-to-learn-c-plus-plus/