**Project 1 Float Number**

Name- Soshi Jain

Professor- Guozhen An

Course- ET 580

**Project Description: FloatNumber Class**

**Introduction**

The **FloatNumber** class is a way to work with numbers that have decimal parts. Instead of using normal floating-point numbers, this class keeps the integer and decimal parts separate. This helps to avoid small mistakes that can happen with floating-point calculations.

**Purpose**

This program is made to do calculations with decimal numbers more accurately. It lets you add, subtract, multiply, and divide numbers while keeping the decimal values precise. You can also compare numbers and use input and output functions easily.

**Code explanation**

The **FloatNumber** class has two private variables:

**-integer**: Stores the whole number part.

**-decimal**: Stores the decimal part separately.

**Constructors (Creating Objects)**

**-Default Constructor**: Sets both integer and decimal to zero.

**-Constructor with Double**: Takes a decimal number and splits it into an integer and decimal part.

**-Constructor with Integer and Decimal**: Lets you give the integer and decimal parts separately.

**Accesor and Mutator Functions**

**getInteger()** and **getDecimal()** return the integer and decimal parts.

**setInteger(i)** and **setDecimal(d)** allow changing the values.

**Arithmetic Operations**

**-operator+()**: Adds two FloatNumber objects and adjusts the decimal part if needed.

**-operator-()**: Subtracts one FloatNumber from another and fixes negative decimals.

**-operator\*():** Multiplies two FloatNumbers and converts the result.

**-operator/()**: Divides two FloatNumbers and checks for division by zero.

**Comparison Operators**

**>** Checks if the left number is greater than the right number.

**<** Checks if the left number is less than the right number.

**>=** Checks if the left number is greater than or equal to the right number.

**<=** Checks if the left number is less than or equal to the right number.

**==** Checks if both numbers are equal.

**!=** Checks if both numbers are not equal.

**Operator Math Operations**

**+=** Adds the right-hand operand to the left-hand operand and assigns the result to the left-hand operand.

**-=** Subtracts the right-hand operand from the left-hand operand and assigns the result to the left-hand operand.

**\*=** Multiplies the left-hand operand by the right-hand operand and assigns the result to the left-hand operand.

**/=** Divides the left-hand operand by the right-hand operand and assigns the result to the left-hand operand.

**Increment and Decrement**

**++** prefix and postfix: adds 1 to the integer part.

**--** prefix and postfix: subtracts 1 from the integer part.

**Using Indexes**

**-operator[]** lets you access the integer (index 0) or decimal (index 1) part.

**Converting to Normal Numbers**

**-toDouble()** changes the FloatNumber object into a normal double value.

**Input and Output**

**-operator>>**: Lets you enter a number by typing the integer and decimal parts separately.

**-operator<<**: Displays the number in an easy-to-read format.

**How to Use It**

**How to Compile**

Use the **Makefile** to build the program:

mingw-32 make all

./prog

To clean up files:

make clean

To use this program, compile and run **Main.cpp**. You can create FloatNumber objects and perform math operations on them. The program also allows comparing numbers and increasing or decreasing their values.