CSC2111 Computer Science I Lab

Lab 17

Objectives

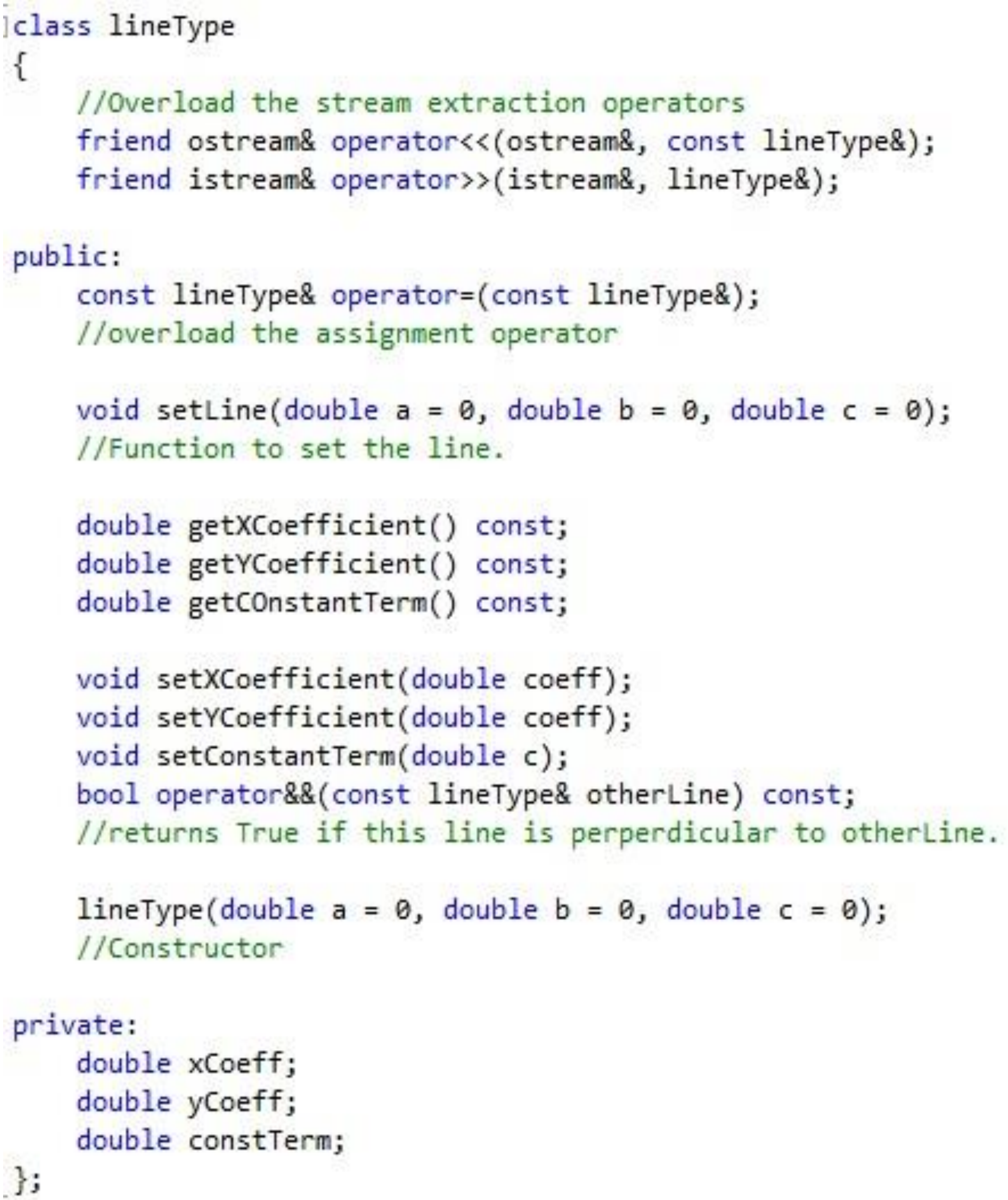
Operator Overloading

Example

- Write a program to overload ***&&*** operator for *lineType* class to check if two lines are perpendicular lines.(Line formula aX+bY=c)

Math background:
-Perpendicular Lines:
Two lines are perpendicular when one is horizontal and the other vertical, or when the product of their slopes is −1.

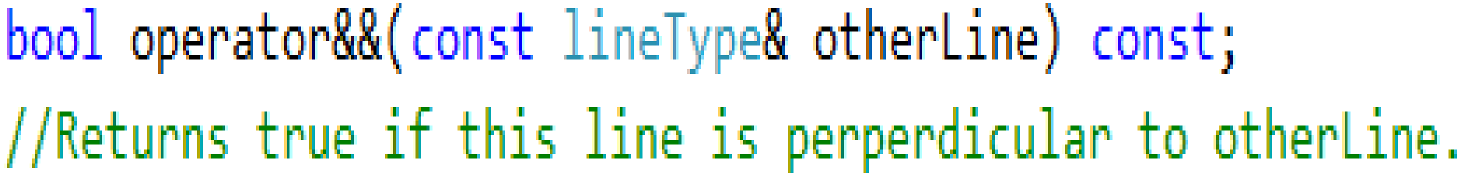
lineType declaration



```
class lineType
{
    //Overload the stream extraction operators
    friend ostream& operator<<(ostream&, const lineType&);
    friend istream& operator>>(istream&, lineType&);

public:
    const lineType& operator=(const lineType&);
    //overload the assignment operator

    void setLine(double a = 0, double b = 0, double c = 0);
    //Function to set the line.

    double getXCoefficient() const;
    double getYCoefficient() const;
    double getCOnstantTerm() const;

    void setXCoefficient(double coeff);
    void setYCoefficient(double coeff);
    void setConstantTerm(double c);
    bool operator&&(const lineType& otherLine) const;
    //returns True if this line is perperdicular to otherLine.

    lineType(double a = 0, double b = 0, double c = 0);
    //Constructor

private:
    double xCoeff;
    double yCoeff;
    double constTerm;
};
```

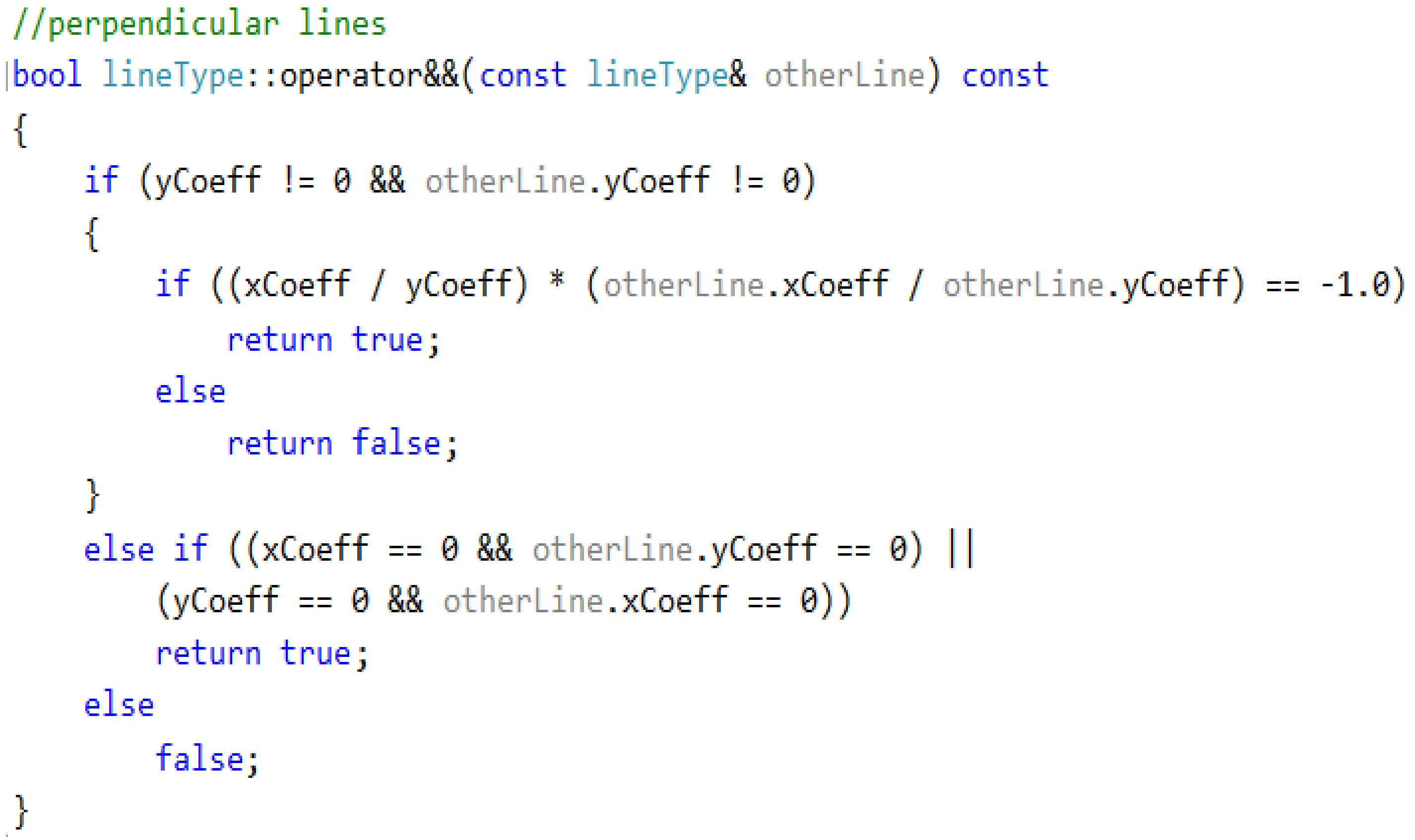
Solution

Add operator function prototype into your class definitions and then complete its definition based on its operation.



```cpp
bool operator&&(const lineType& otherLine) const;
//Returns true if this line is perperdicular to otherLine.
```

Solution



```
//perpendicular lines
bool lineType::operator&&(const lineType& otherLine) const
{
    if (yCoeff != 0 && otherLine.yCoeff != 0)
    {
        if ((xCoeff / yCoeff) * (otherLine.xCoeff / otherLine.yCoeff) == -1.0)
            return true;
        else
            return false;
    }
    else if ((xCoeff == 0 && otherLine.yCoeff == 0) ||
        (yCoeff == 0 && otherLine.xCoeff == 0))
        return true;
    else
        false;
}
```