Q . Operator Overloading

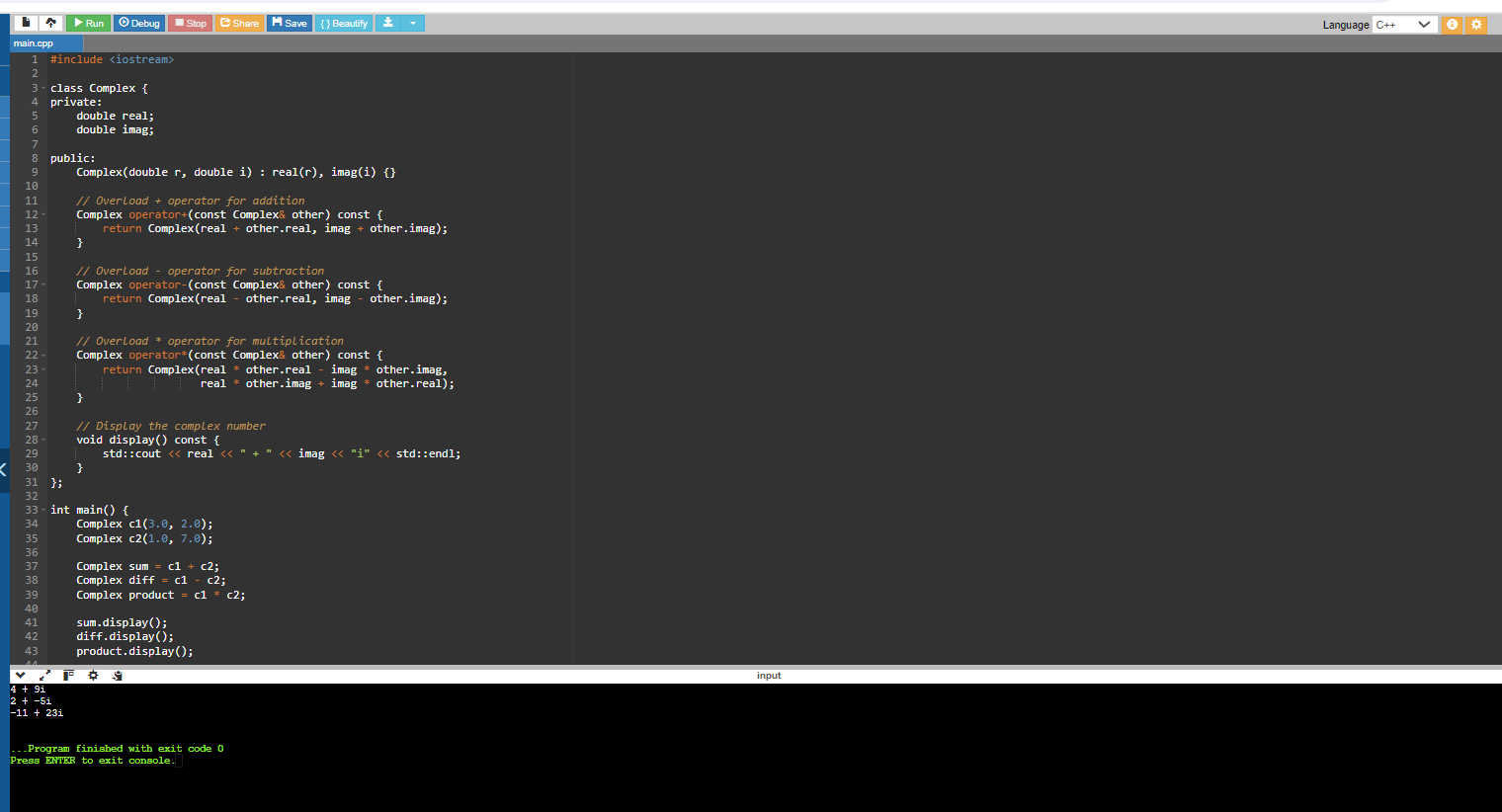
Complex Numbers (C++) - Define a class Complex to represent complex numbers with member variables for real and imaginary parts. Overload the +, -, and \* operators for complex number addition, subtraction, and multiplication.

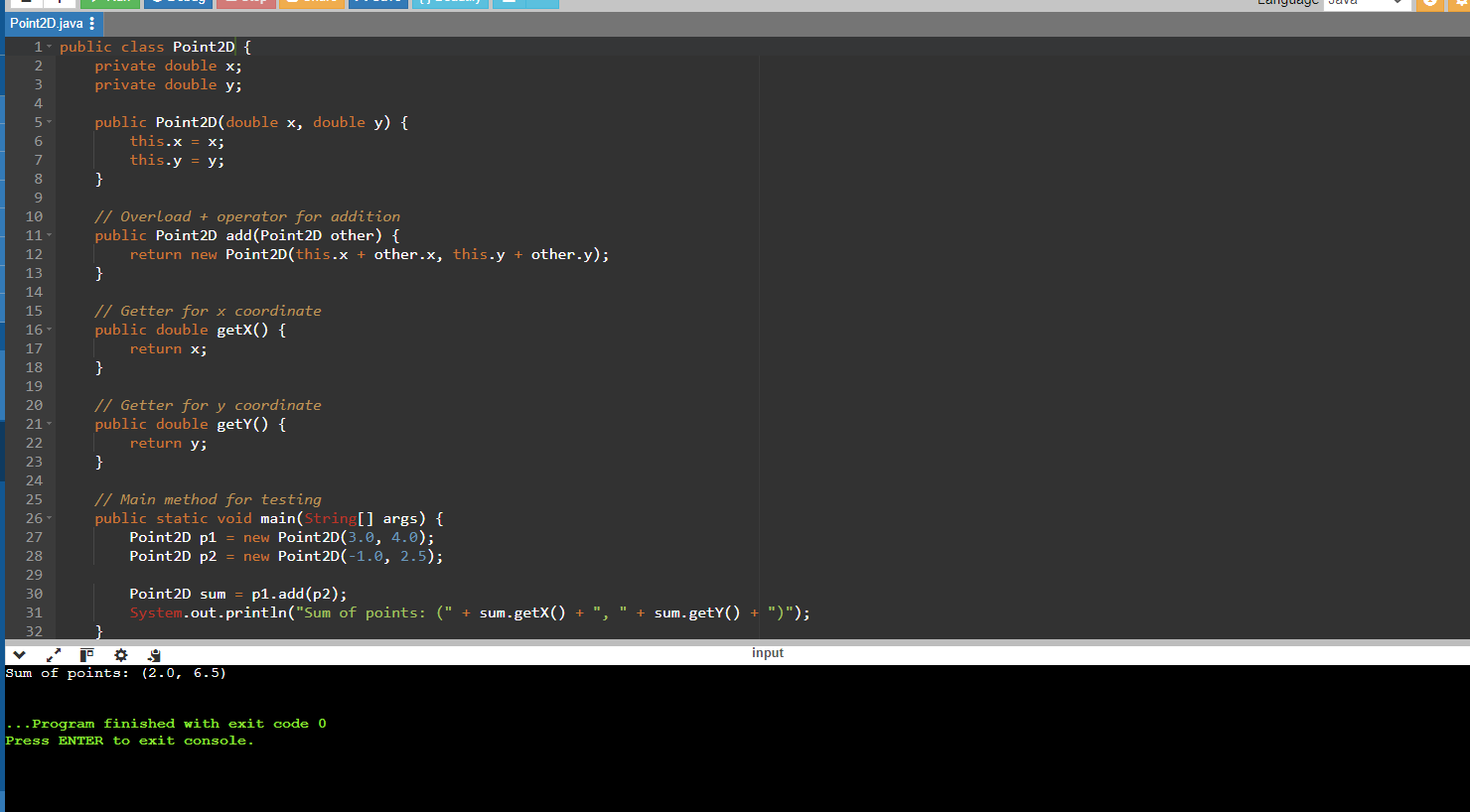
Point2D (Java) - Create a class Point2D with x and y coordinates. Overload the + operator to return a new Point2D object representing the sum of two points.

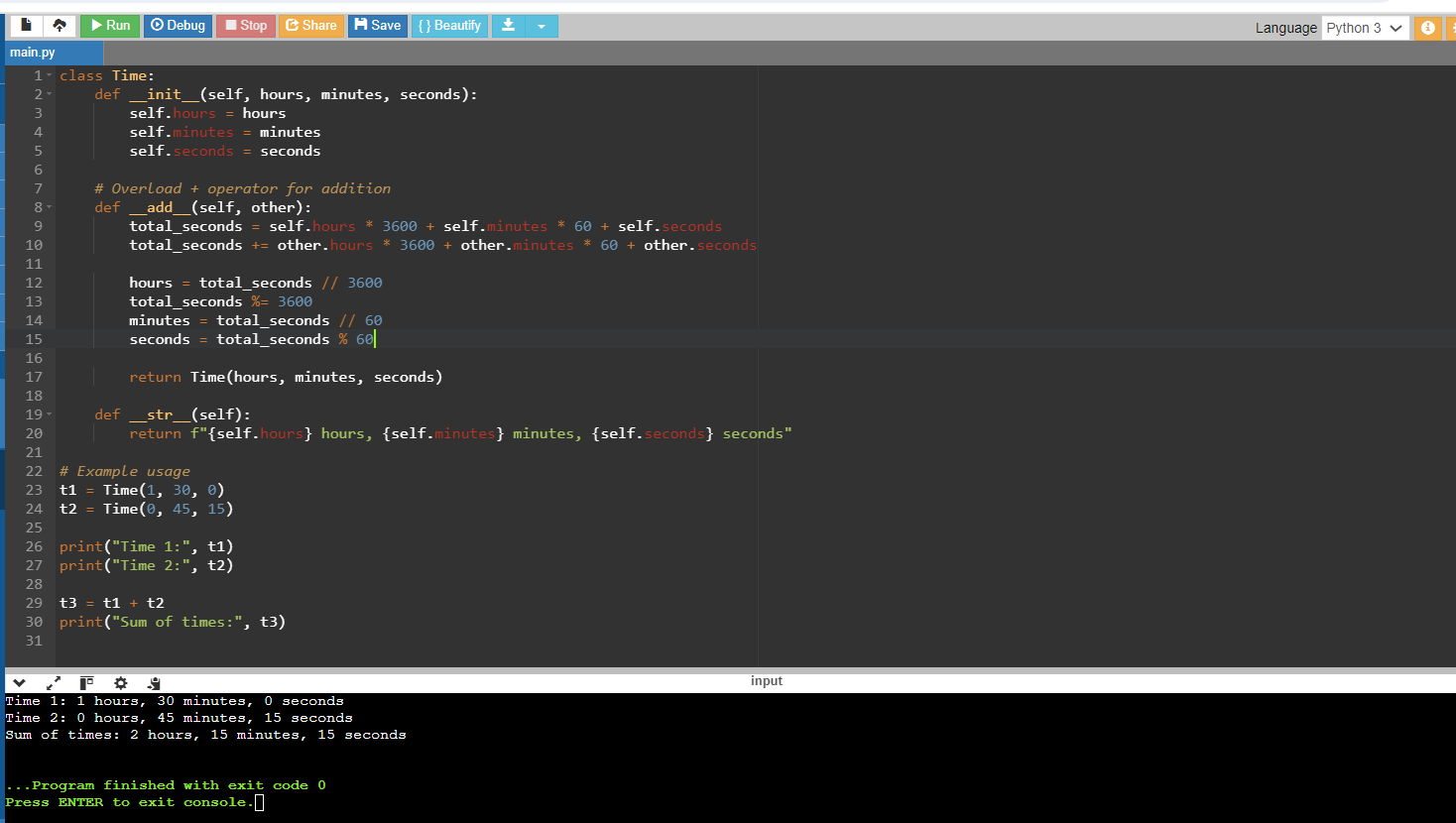
Time (Python) - Design a class Time to store hours, minutes, and seconds. Overload the + operator to add two Time objects and return a new Time object with the combined duration.

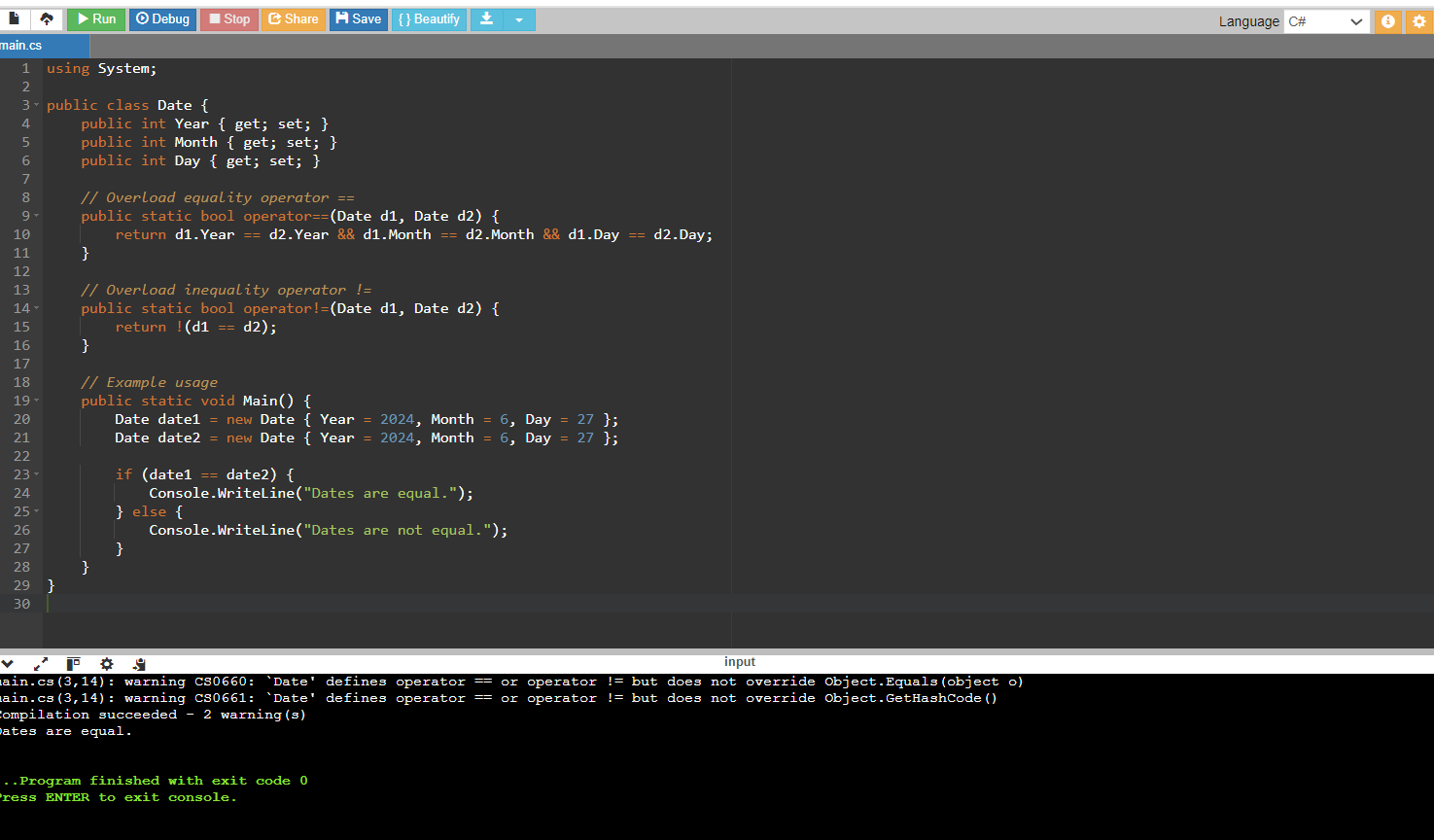
Date (C#) - Implement a class Date with year, month, and day. Overload the comparison operators (== and !=) to compare two Date objects.

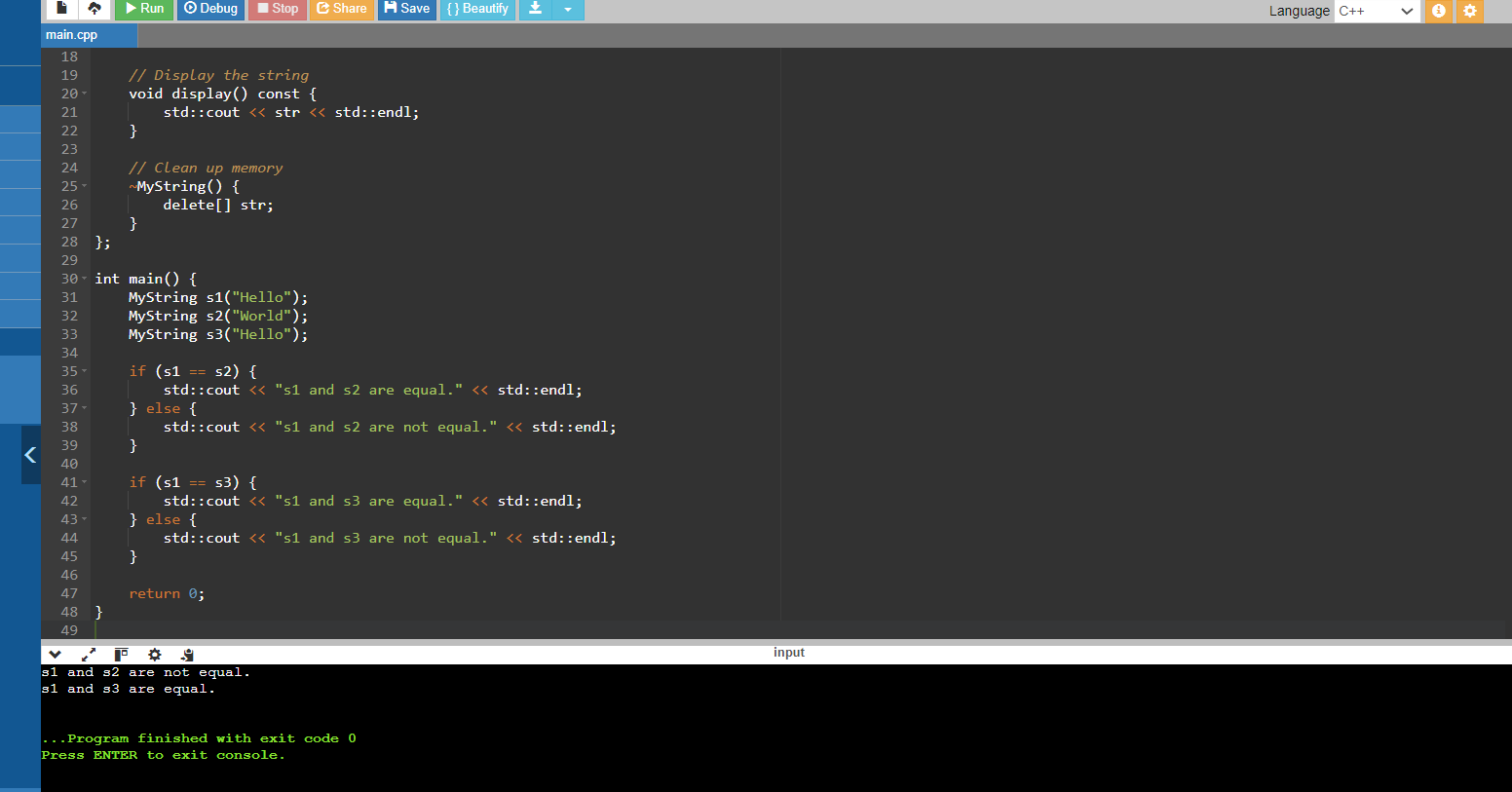
String Equality (C++) - Overload the equality operator (==) for a custom String class to compare string contents (not just memory addresses).











Q . Area Calculation (Java) - Create a function calculateArea that can handle different shapes (e.g., rectangle, circle) by overloading it with parameters like width, height, or radius.

Unit Conversion (Python) - Design a function convert that takes a value and a unit (e.g., meters, feet, Celsius, Fahrenheit) and converts it to another unit using appropriate conversion factors.

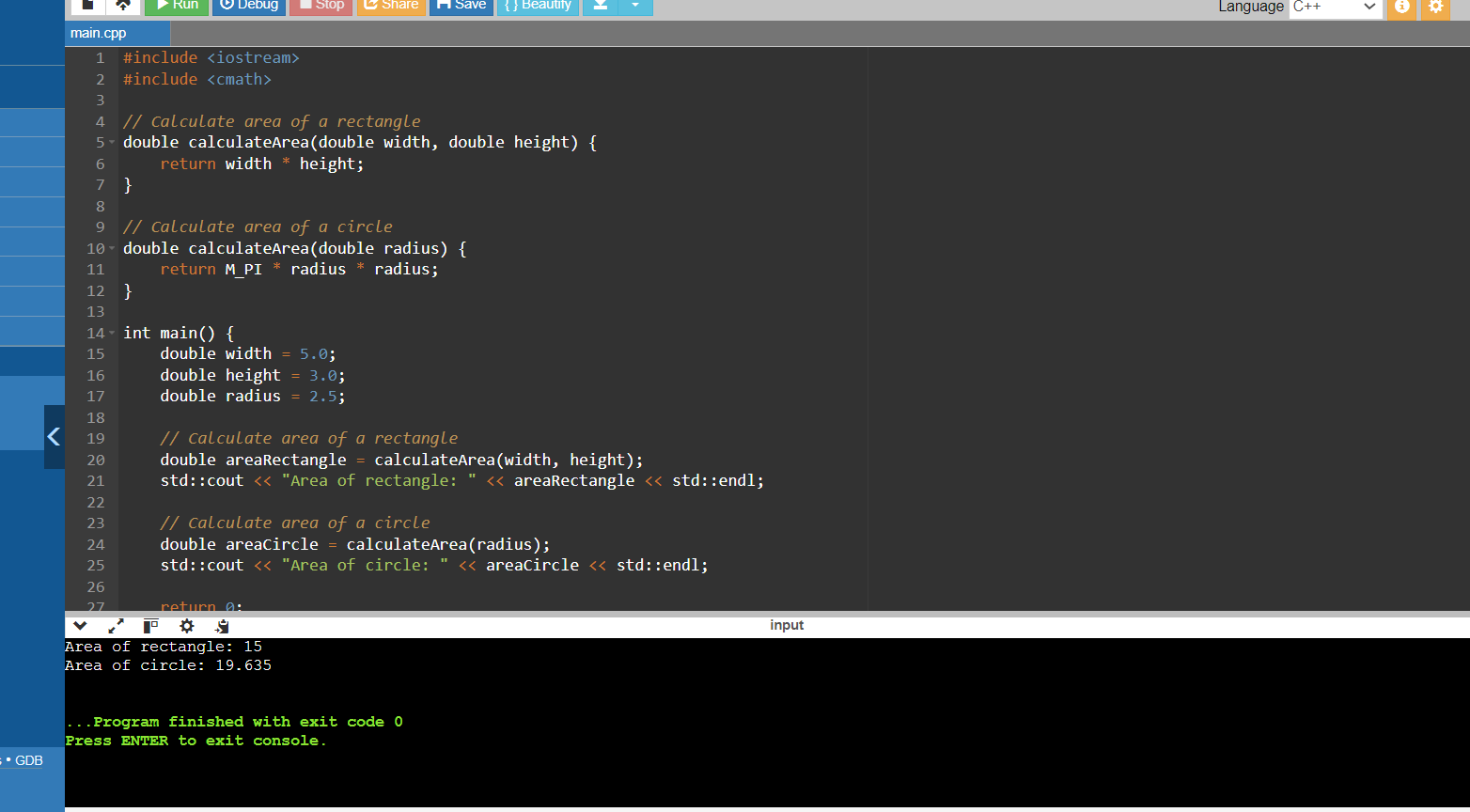
Statistics (C++) - Implement functions average, minimum, and maximum that can take an array of integers or doubles as input, depending on the function call.

String Formatting (C#) - Write overloaded functions formatString that can take a format string and different data types (e.g., int, double, string) to create formatted output strings.

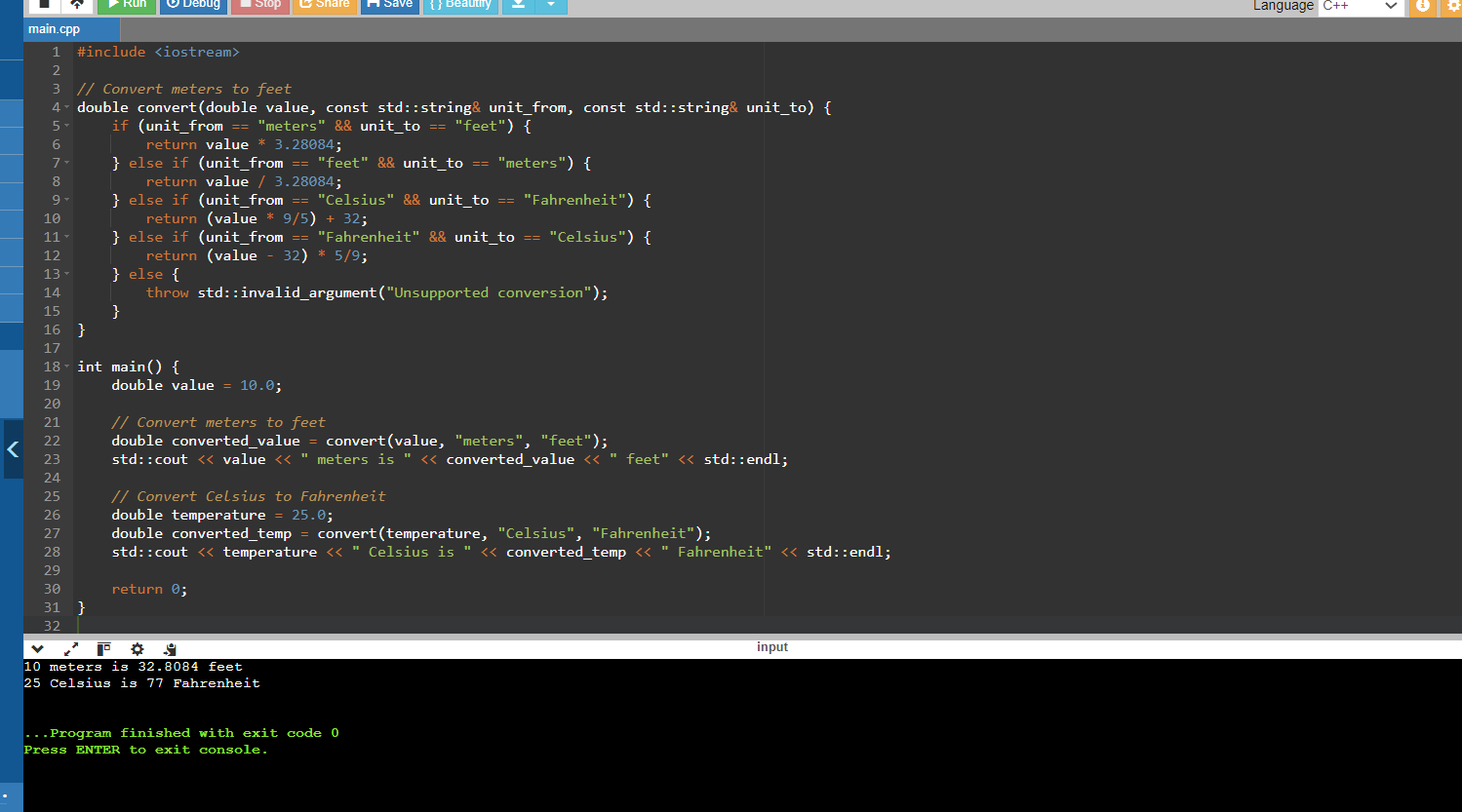
Math Functions (Python) - Create overloaded functions factorial and power that can handle integer and floating-point input for calculating factorials and raising a number to a power.

Combined Concepts

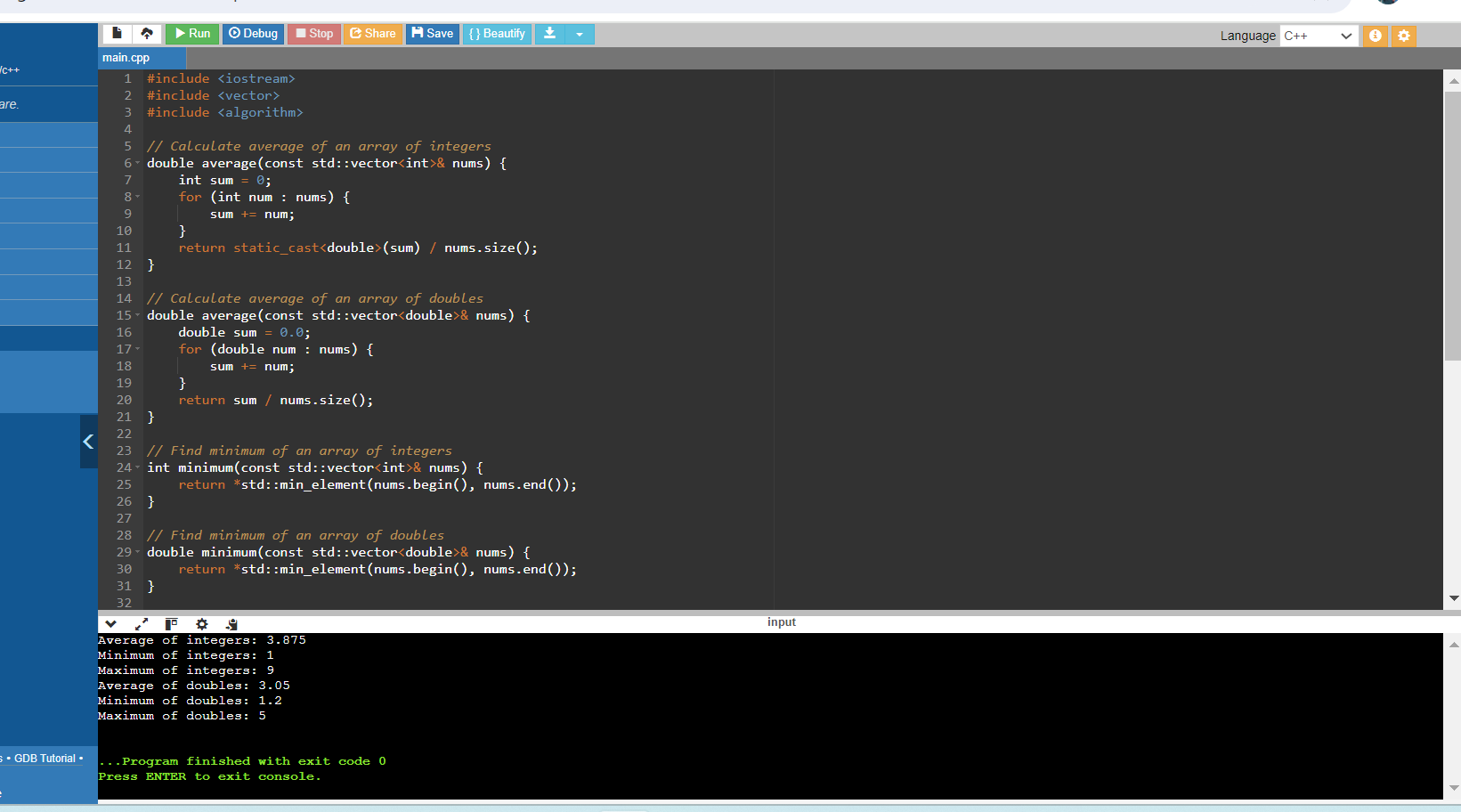
1. Area Calculation (Java)



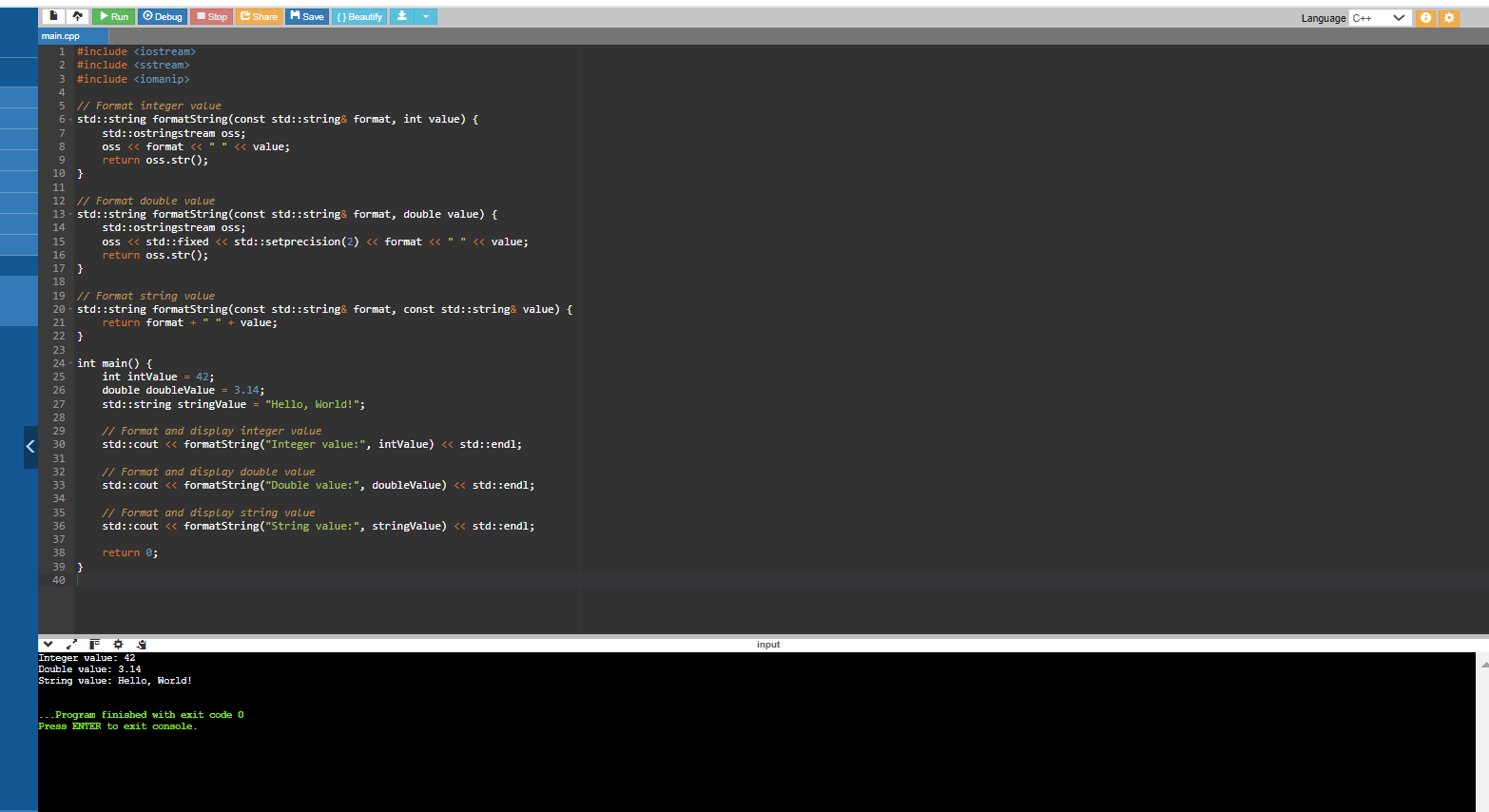
1. Unit Conversion (Python)



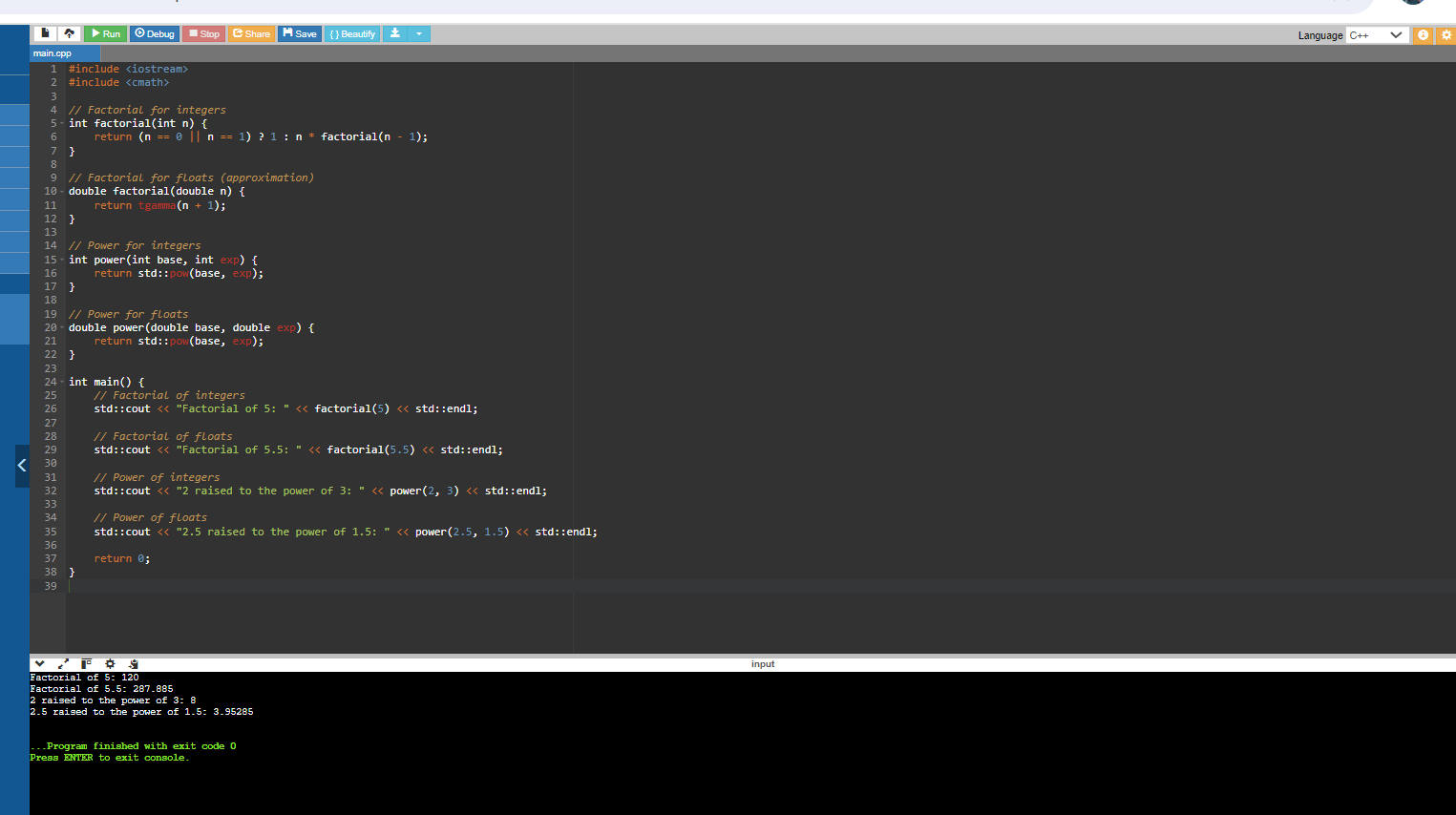
1. **Statistics (C++)**



**4.String Formatting (C#)**



**5.Math Functions (Python)**



**Q. Polynomial Addition (C++) - Define a class Polynomial to represent polynomials with terms (coefficient and exponent). Overload the + operator to add two Polynomial objects and return a new Polynomial with the combined terms.**

**Money Class (Java) - Design a class Money to store currency amount and type (e.g., USD, EUR). Overload the arithmetic operators (+, -, \*, /) for money objects, considering appropriate type conversions and potential rounding issues.**

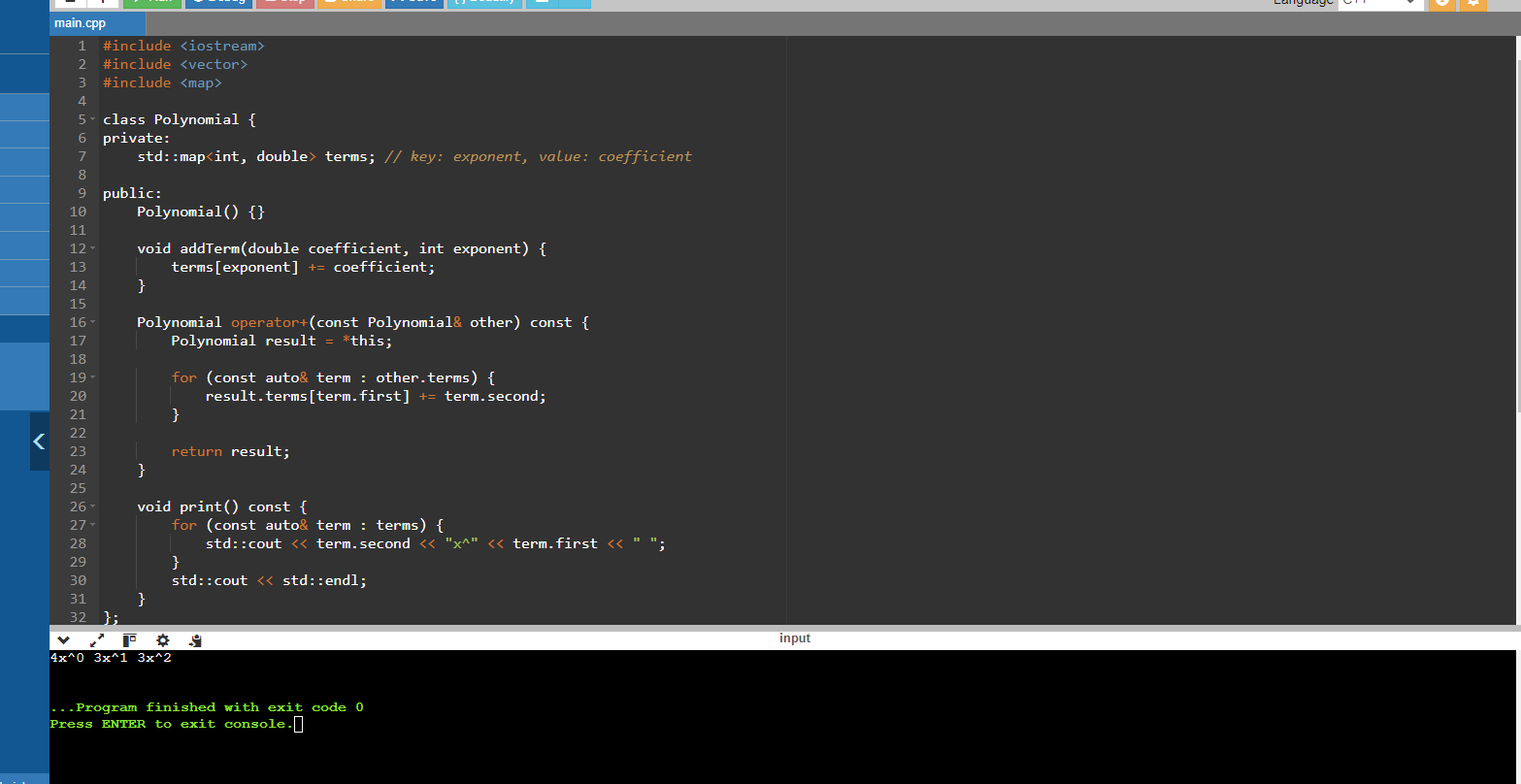
**Shape Hierarchy (Python) - Create a base class Shape with an abstract method getArea. Derive classes like Circle, Rectangle, and Square from Shape and implement the getArea method in each derived class.**

**Inventory Management (C++) - Implement a class Item with properties like name, price, and quantity. Overload the << operator for easy printing of item details to the console.**

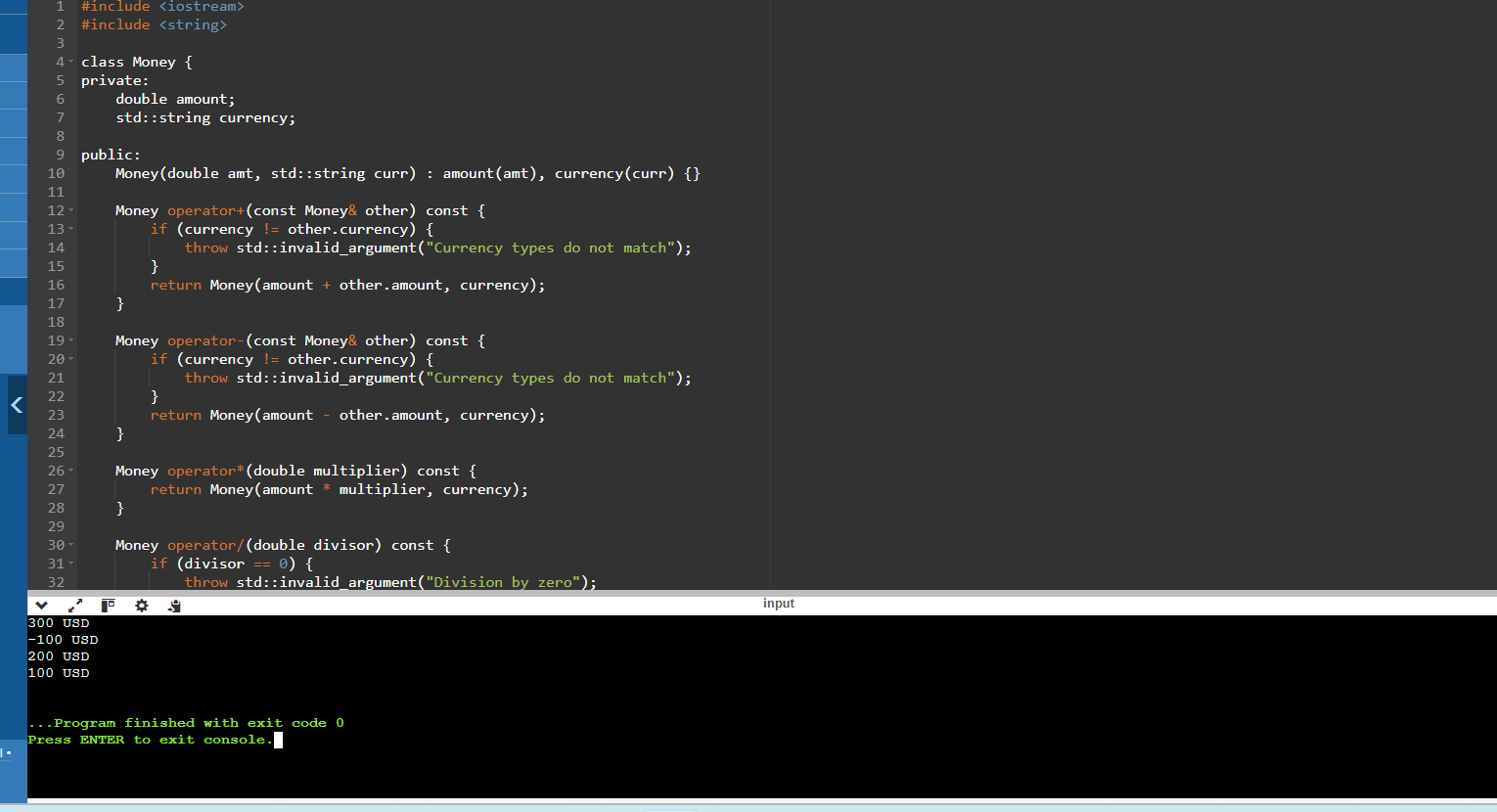
**File I/O with Overloading (C#) - Design a class FileReaderWriter with overloaded functions read and write that can handle different file types (e.g., text, CSV) based on the file extension provided as input.**

**Advanced Challenges**

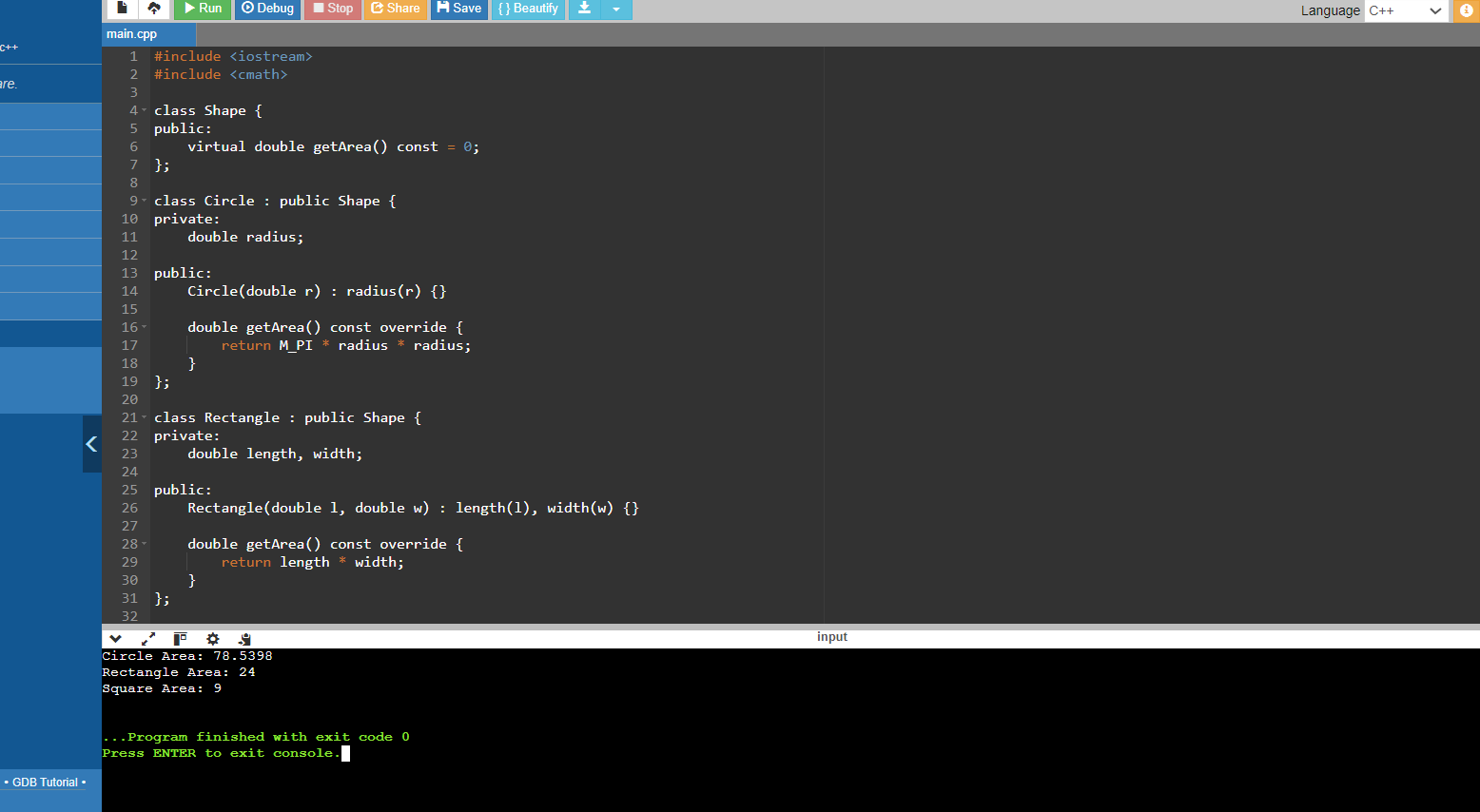
1. Polynomial Addition



2. Money Class



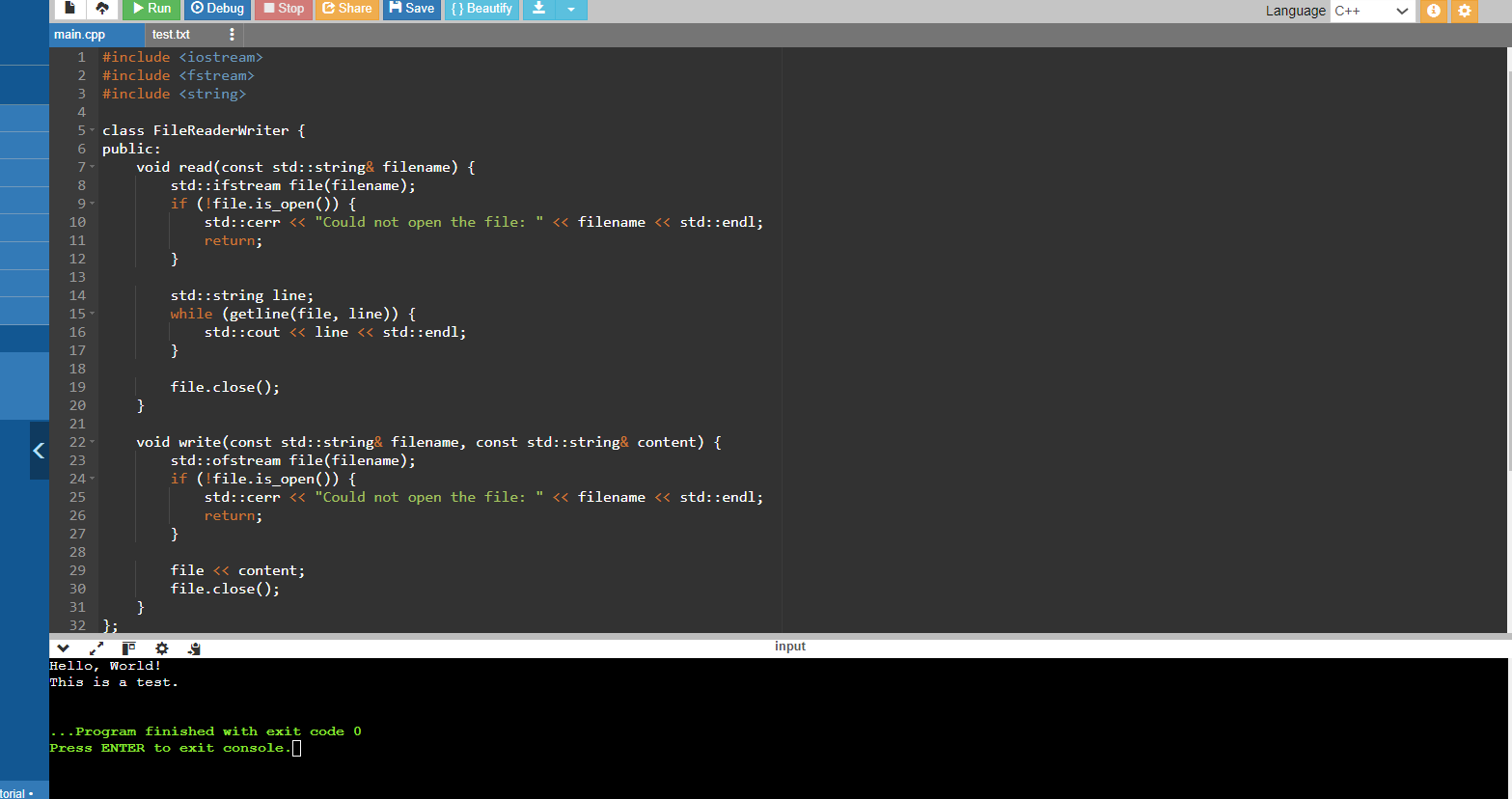
3.Shape Hierarchy



1. Inventory Management



1. File I/O with Overloading



**Q. Matrix Operations (Java) - Create a class Matrix to store a 2D array and overload arithmetic operators (+, -, \*) for matrix addition, subtraction, and multiplication (considering matrix dimensions).**

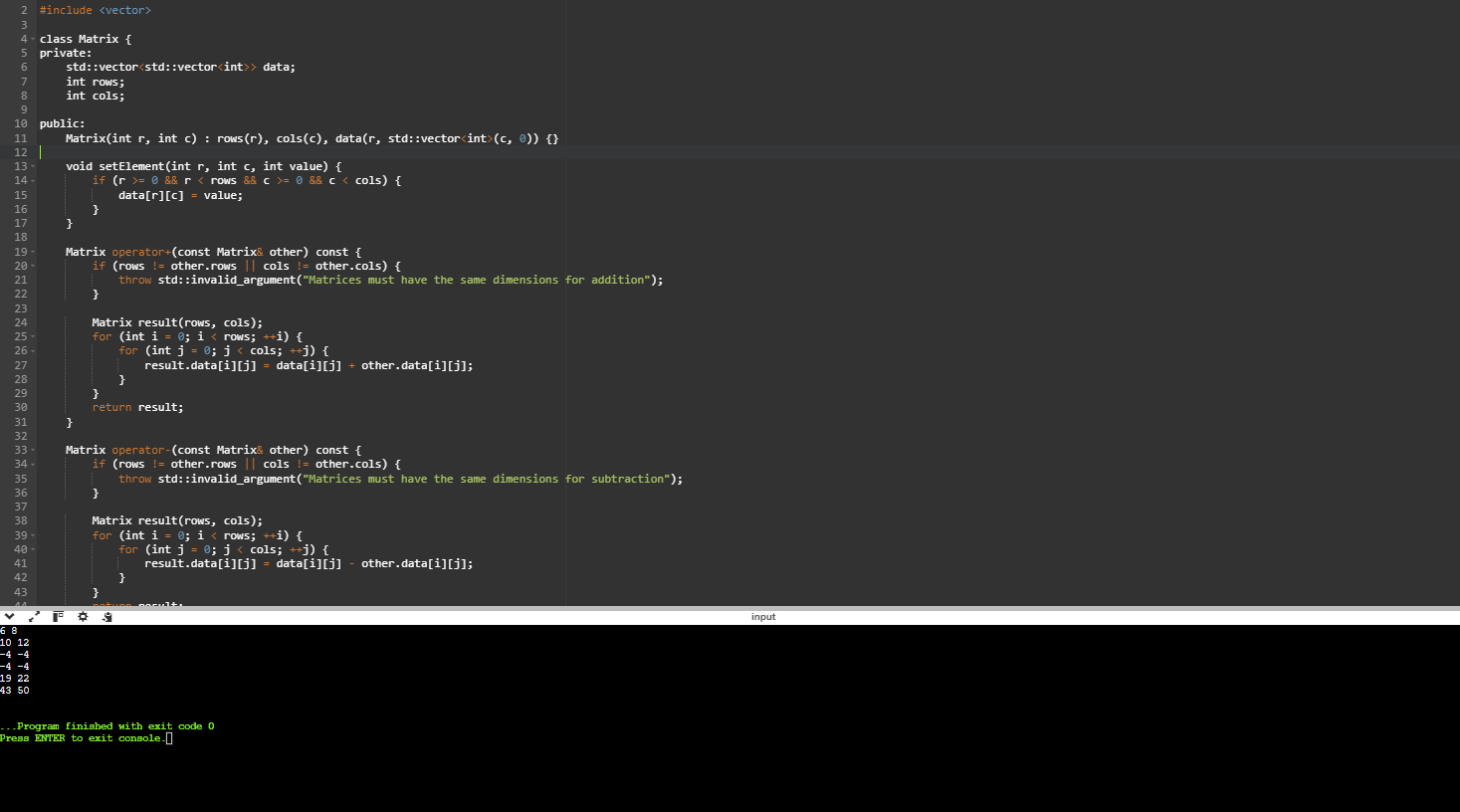
**Custom Container (Python) - Implement a class CustomList that behaves like a list but overloads the subscript operator ([]) to perform boundary checking and prevent out-of-bounds access.**

**Smart Pointers (C++) - Design a smart pointer class MySmartPtr that overloads the dereference operator (\*) and arrow operator (->) for memory management and safe access to the pointed-to object.**

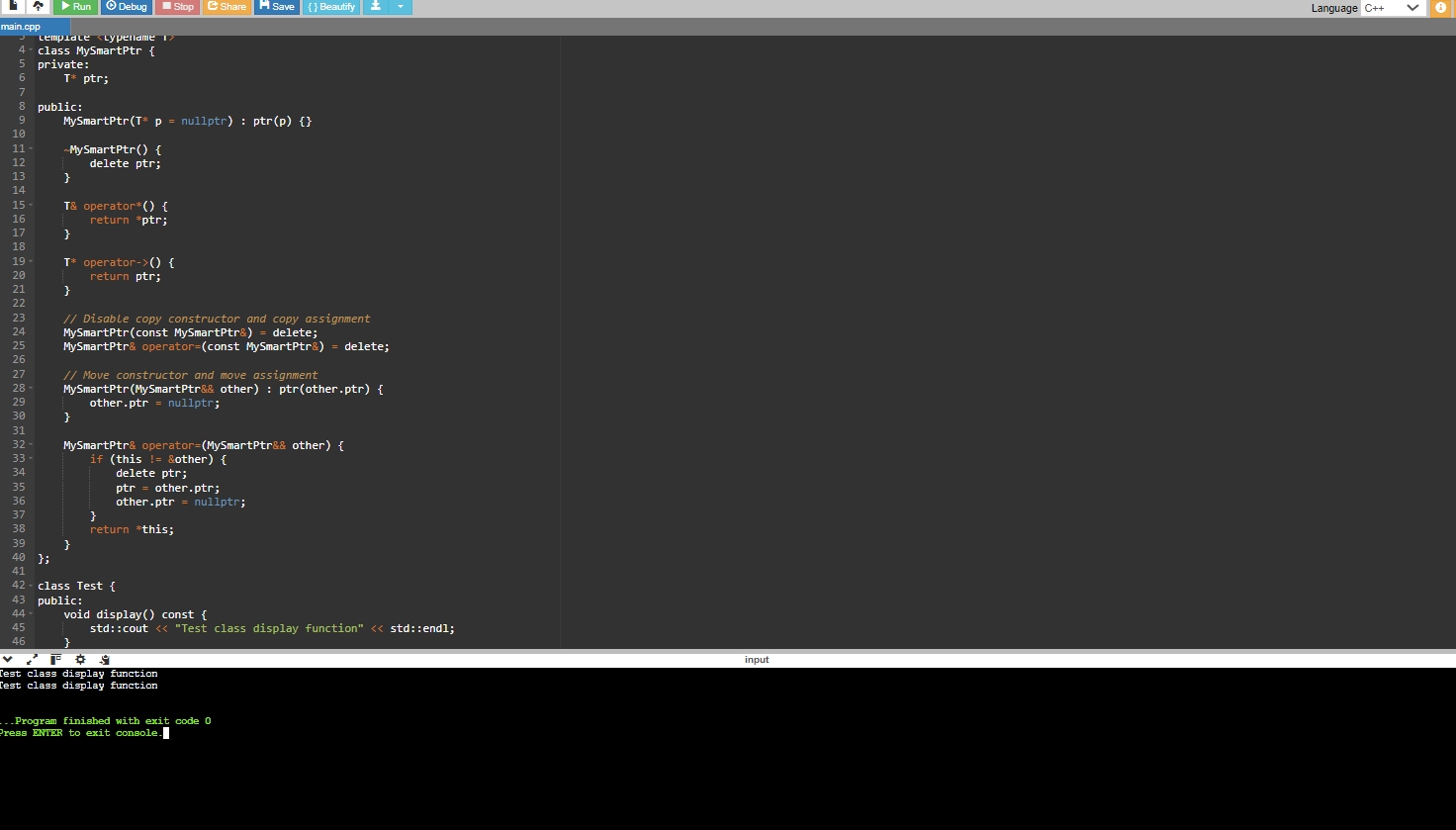
**Custom Stream (C#) - Create a class CharacterStream that overloads the << operator to provide a custom way of writing characters to a stream, applying transformations or filters if needed.**

**Template Classes (C++) - Implement a template class Vector that can store elements of any data type and overload operators (+, -, []) to work with vectors of different types.**

1. Matrix Operations



1. Smart Pointers



1. Template Classes

