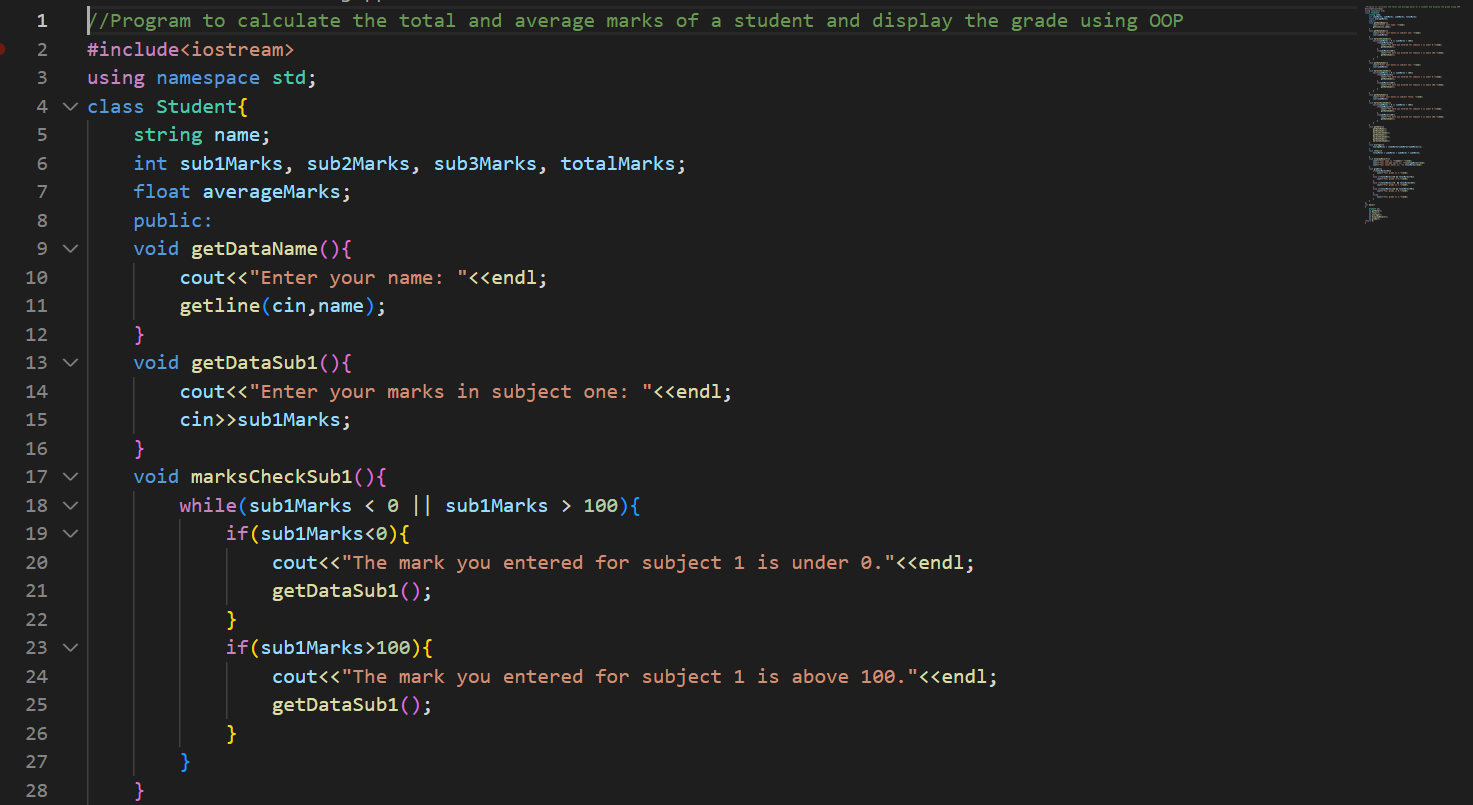
**Task 1: Basic student grading system prototype using classes and objects. [20 Marks]**

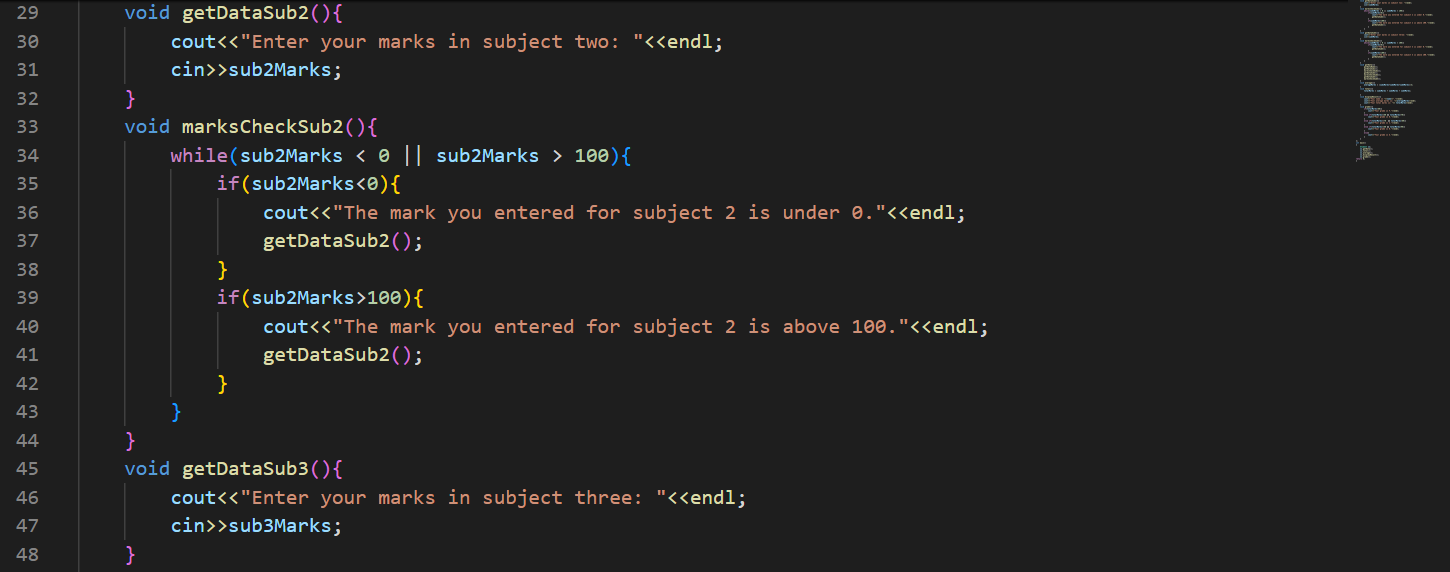
Write a program that manages a simple student grade calculator with the following requirements. Create a Student class that has:

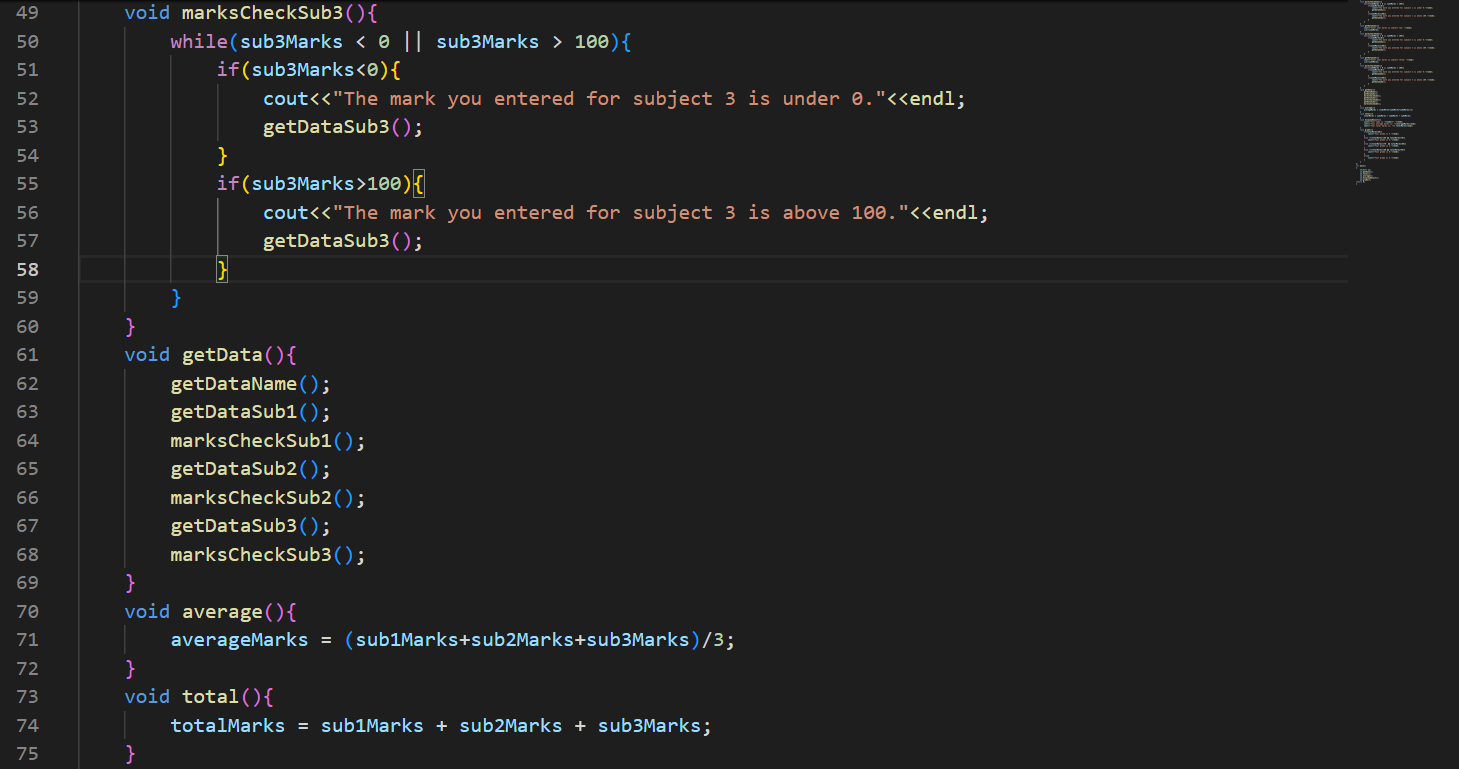
1. Student name (string)
2. Three subject marks (integers)
3. A basic member function to calculate average

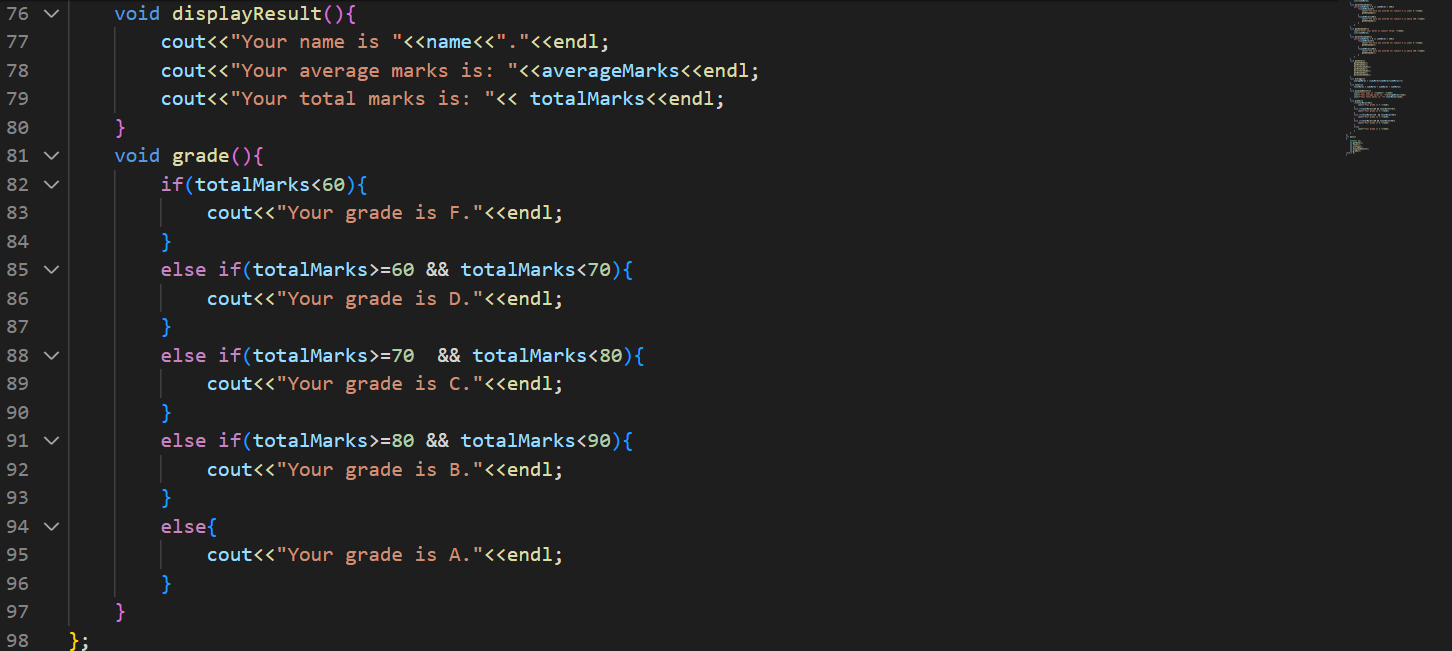
The program should:

1. Accept student details (name and marks) from user input
2. Calculate and display:
   1. Total marks
   2. Average marks
   3. Grade (A for ≥90%, B for ≥80%, C for ≥70%, D for ≥60%, F for <60%)
3. Display a message if any mark is below 0 or above 100

****

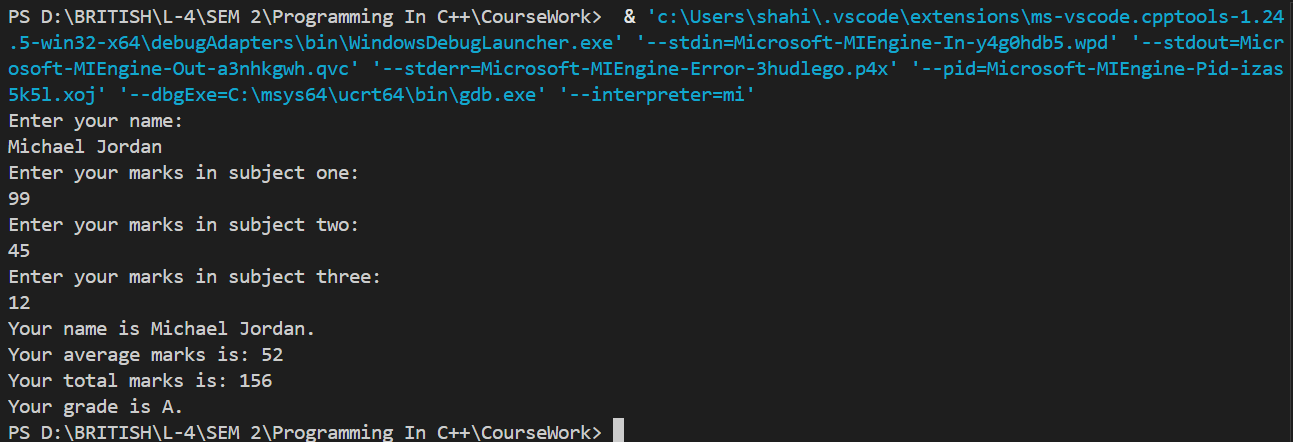
****

****

****

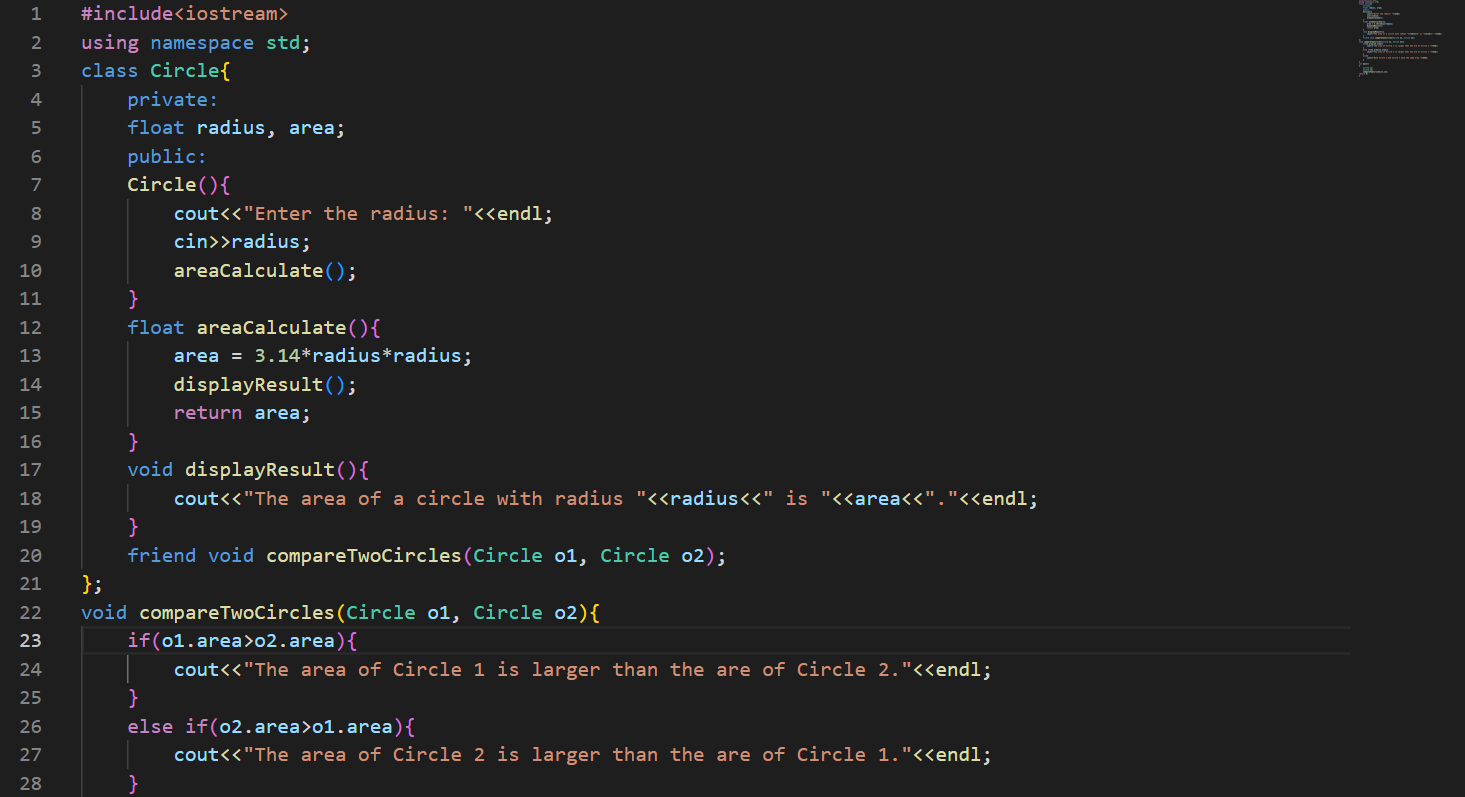
****

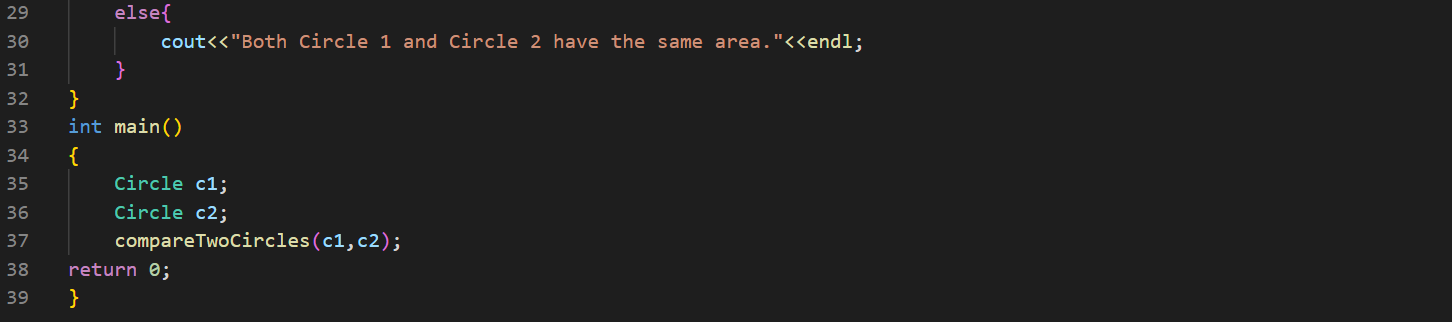
**Solution:**

****

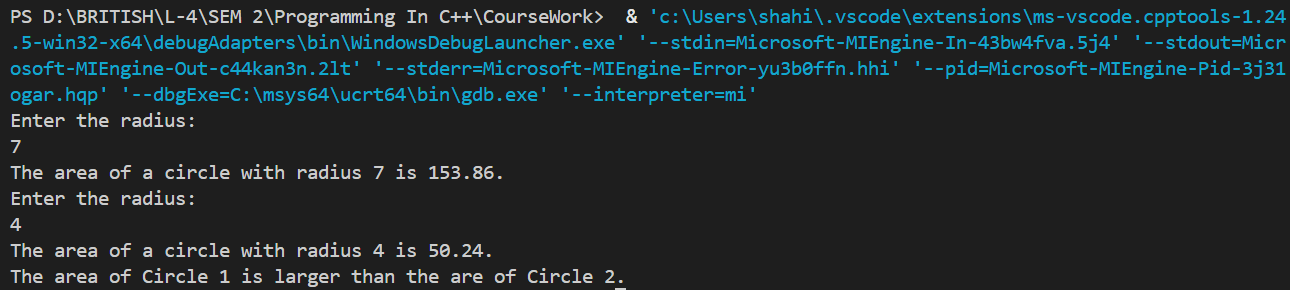
**Task 2: Programming assignments: All questions are mandatory**

1. Write a program with a class Circle having:
   1. Private member: radius (float)
   2. A constructor to initialize radius
   3. A friend function compareTwoCircles that takes two Circle objects and prints which circle has the larger area

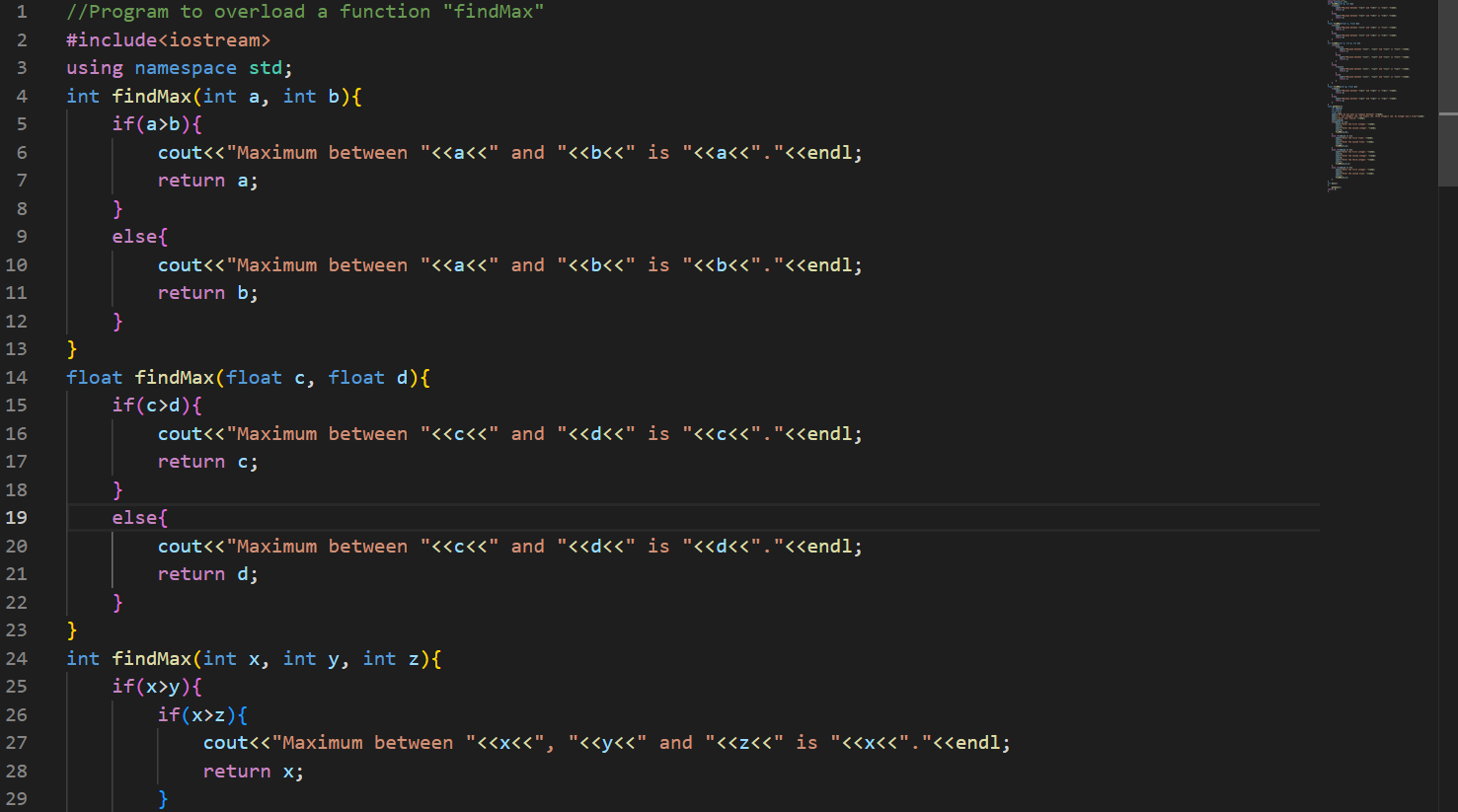
****

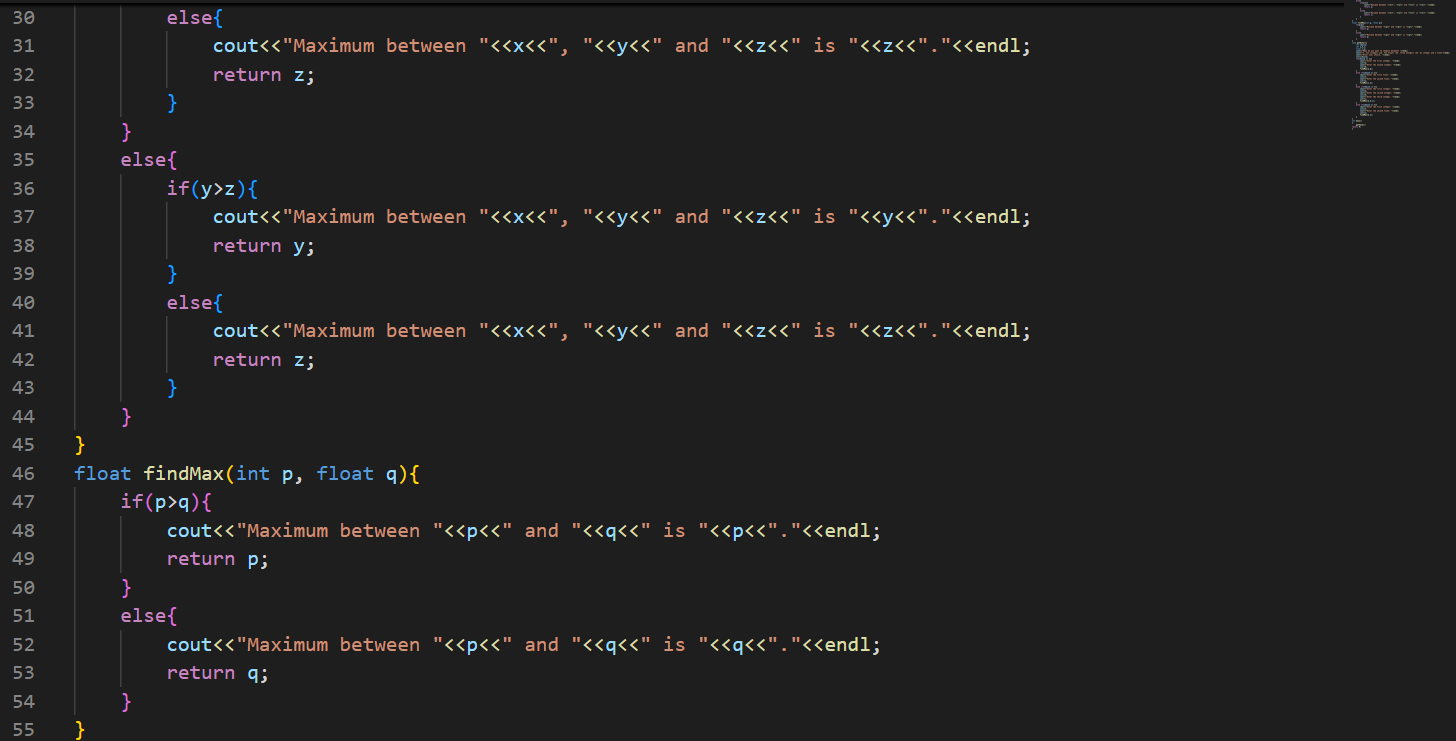
****

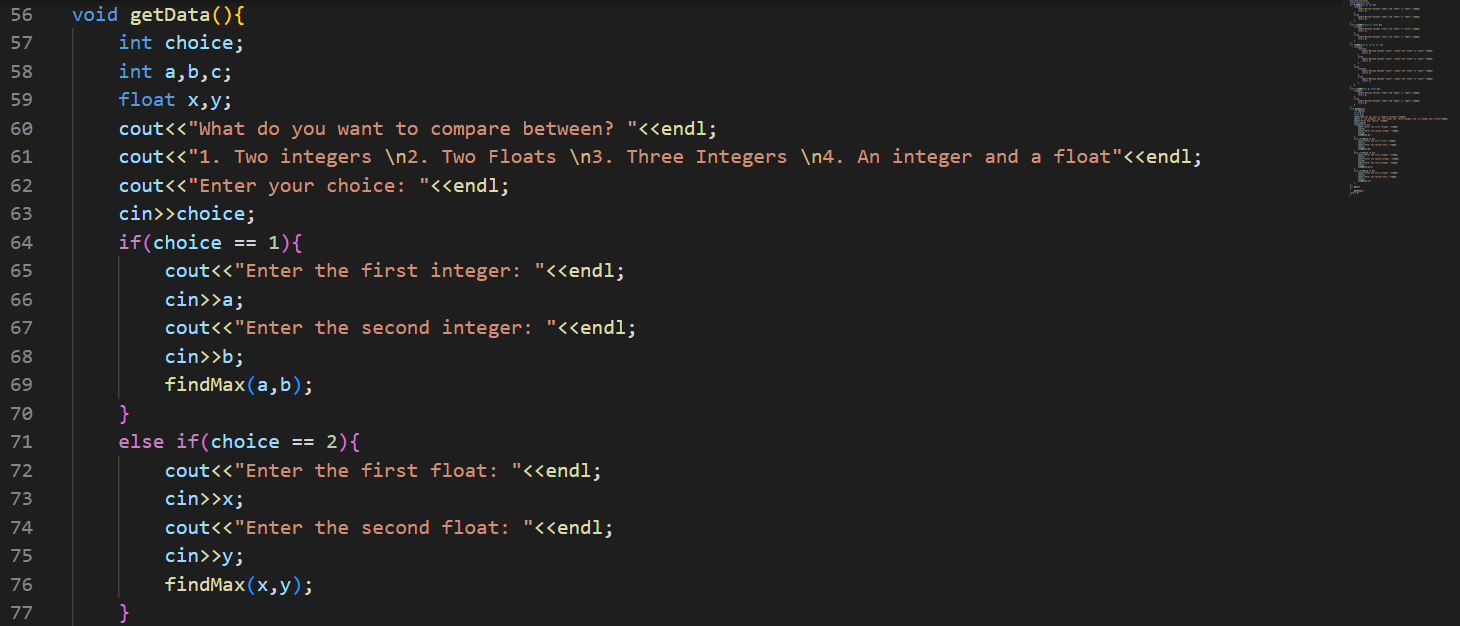
**Solution:**

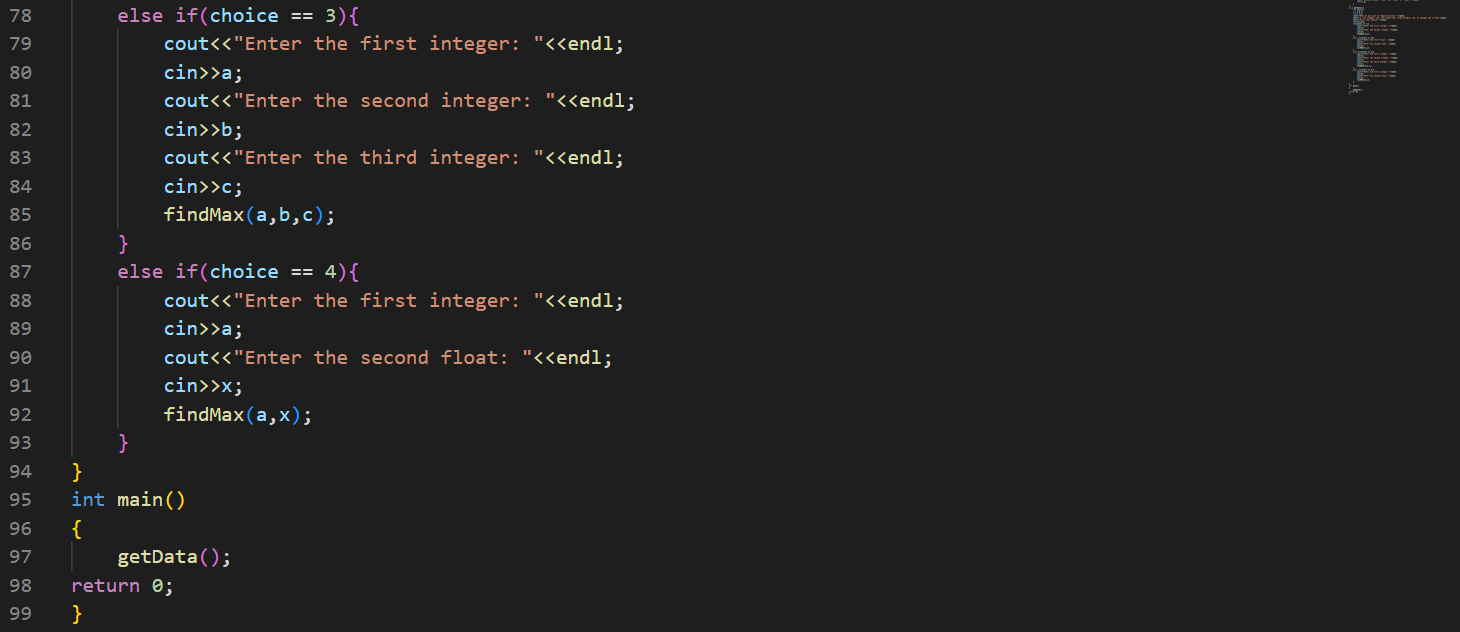
****

1. Create a program with these overloaded functions named findMax:
   1. One that finds maximum between two integers
   2. One that finds maximum between two floating-point numbers
   3. One that finds maximum among three integers
   4. One that finds maximum between an integer and a float **[30 marks]**

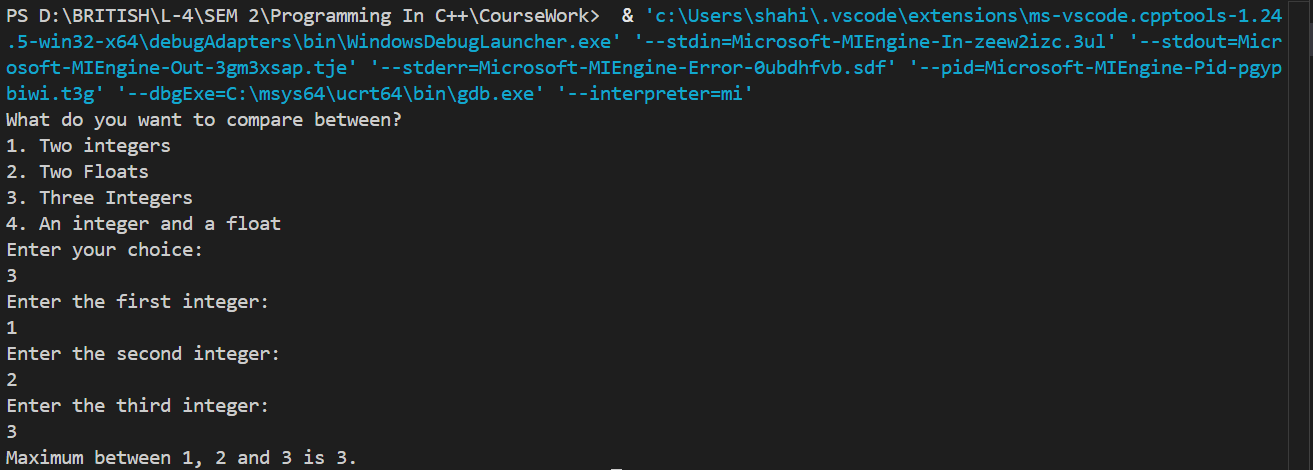
****

****

****

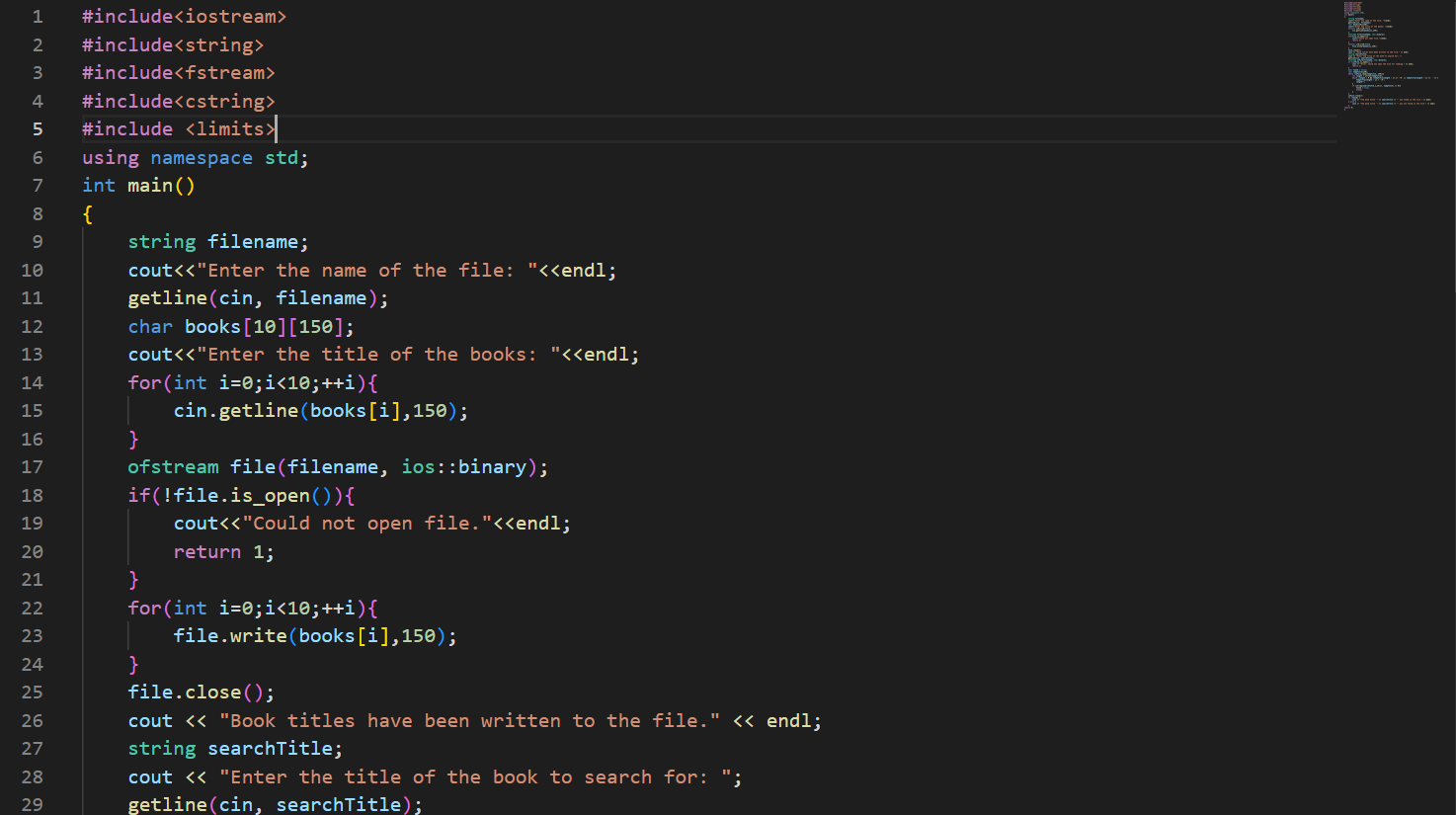
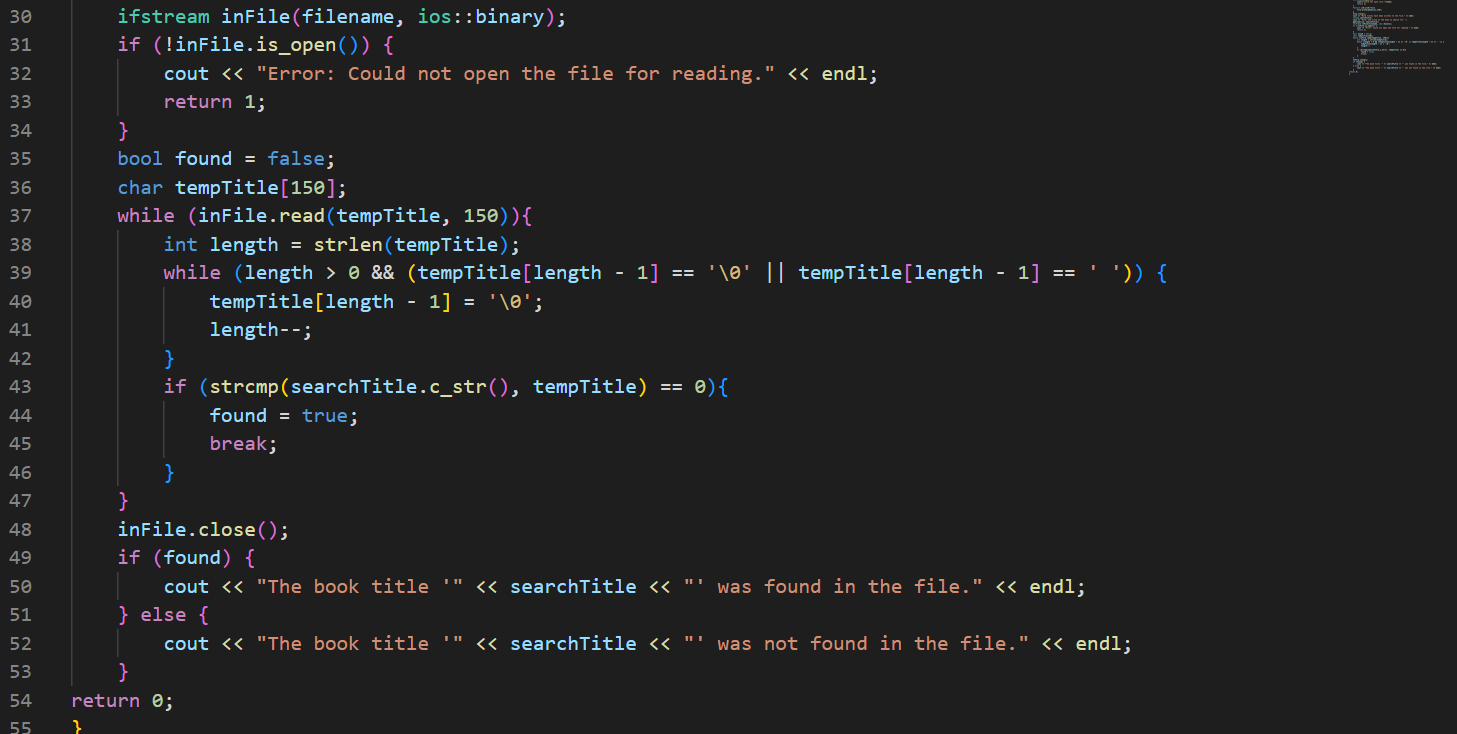
****

**Solution:**

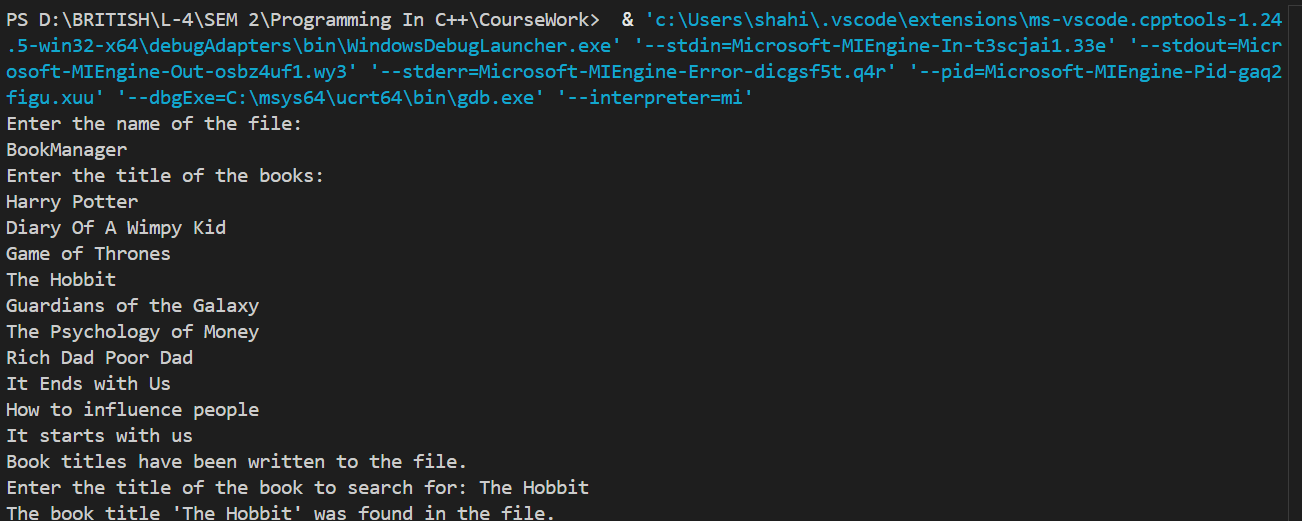
****

**Task 3: Basics of File Handling**

Write a program that reads the titles of 10 books (use an array of 150 characters) and writes them in a binary file selected by the user. The program should read a title and display a message to indicate if it is contained in the file or not.

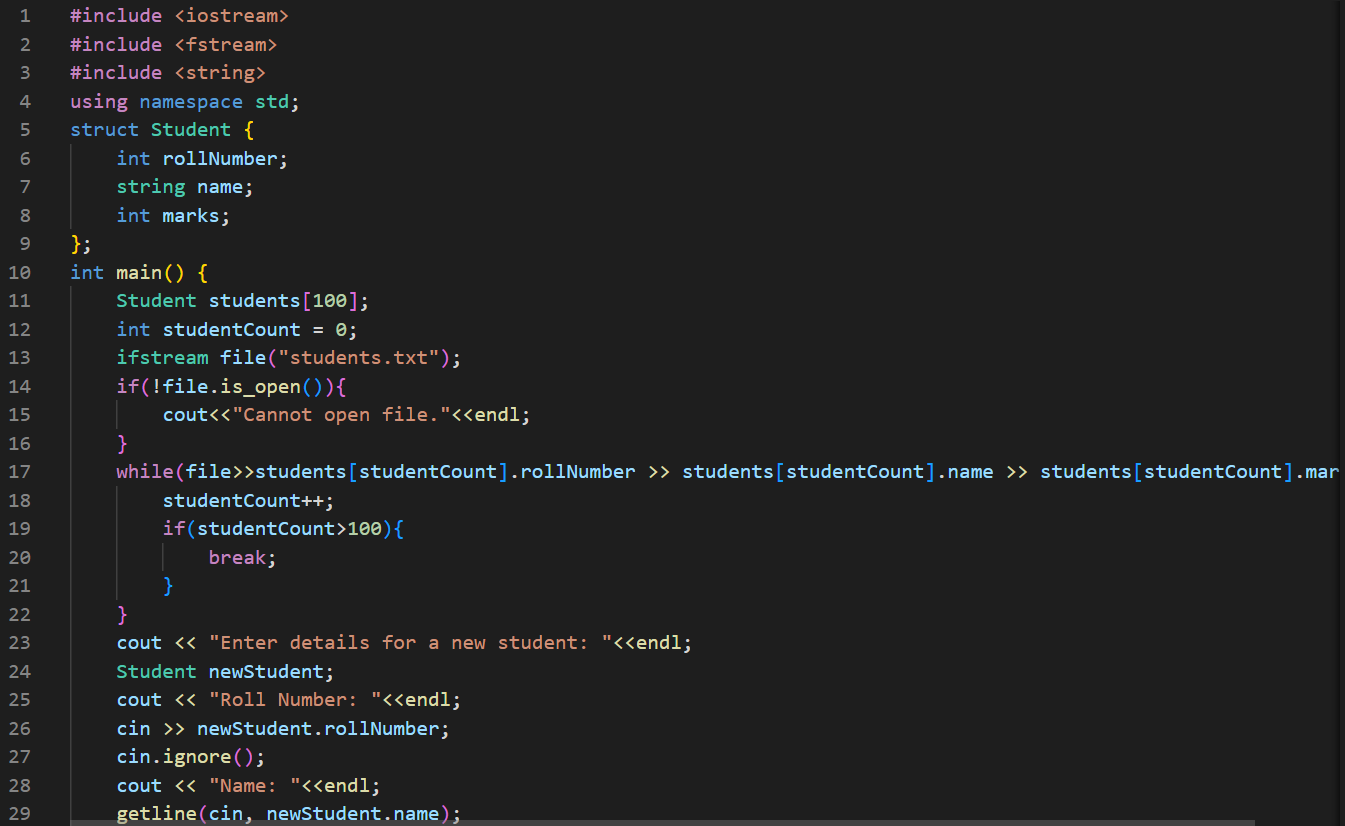
**** ****

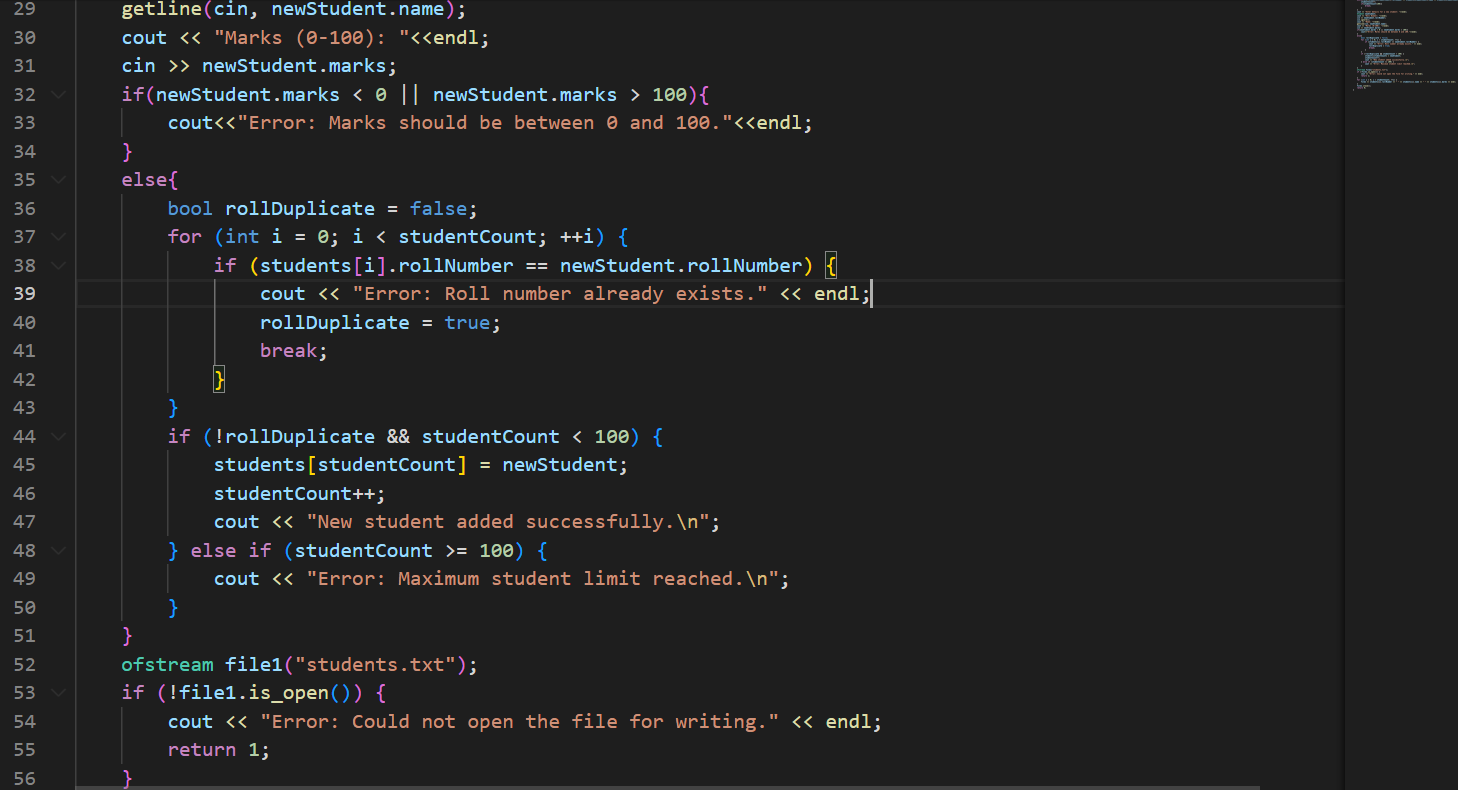
**Solution:**

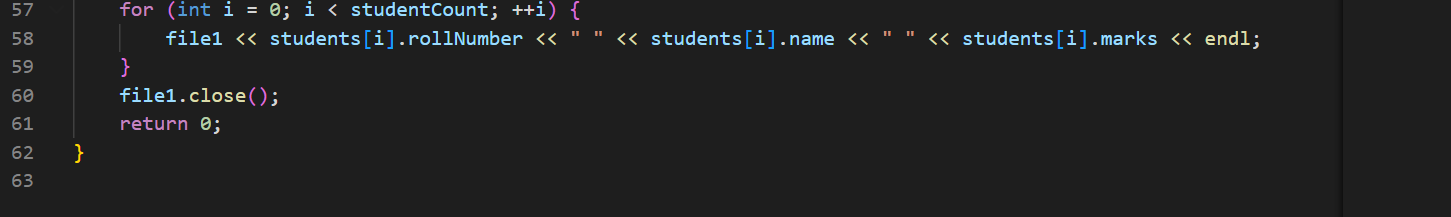
****

Create a program that:

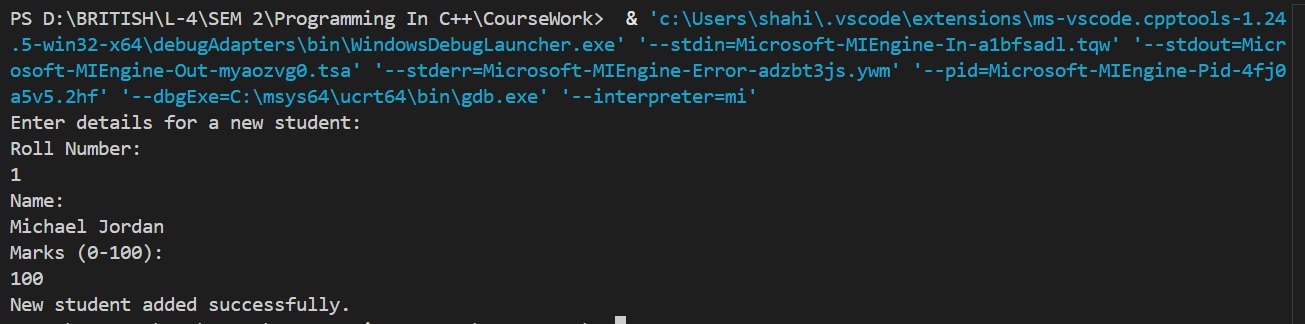
1. Reads student records (roll, name, marks) from a text file
2. Throws an exception if marks are not between 0 and 100
3. Allows adding new records with proper validation
4. Saves modified records back to file







**Solution:**

****

