

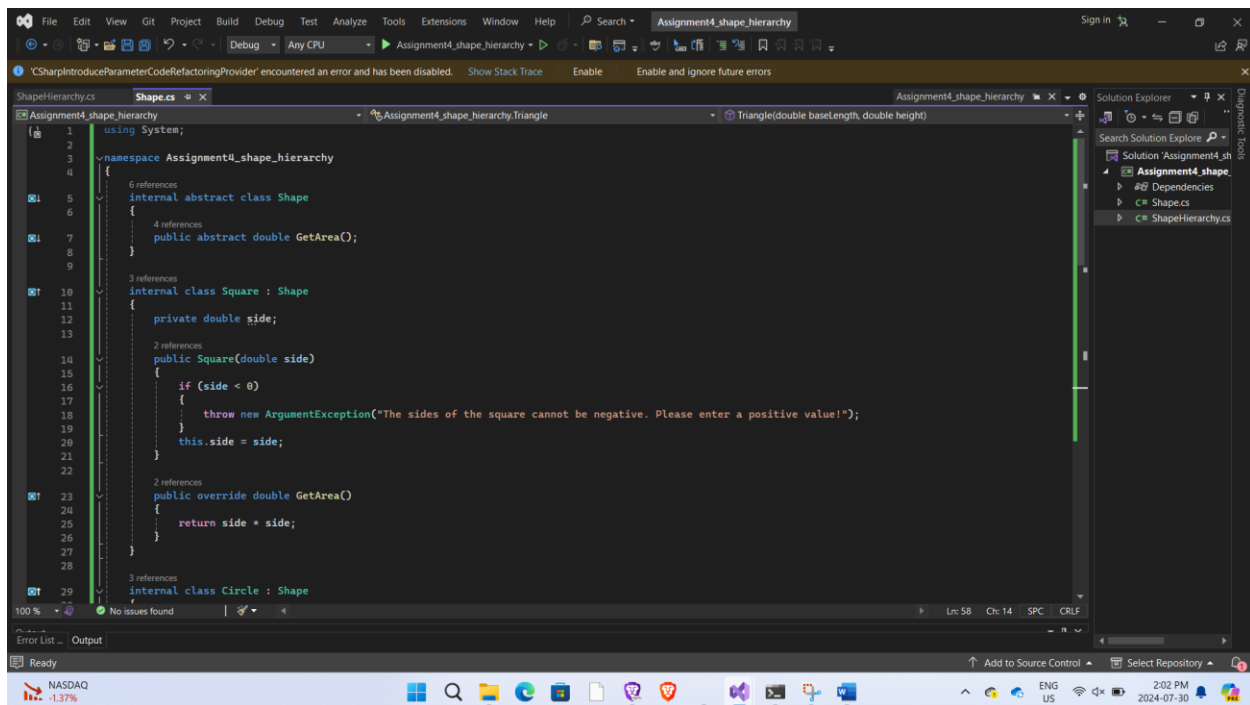
## Assignment #4

Student Name : Shreejana Shrestha

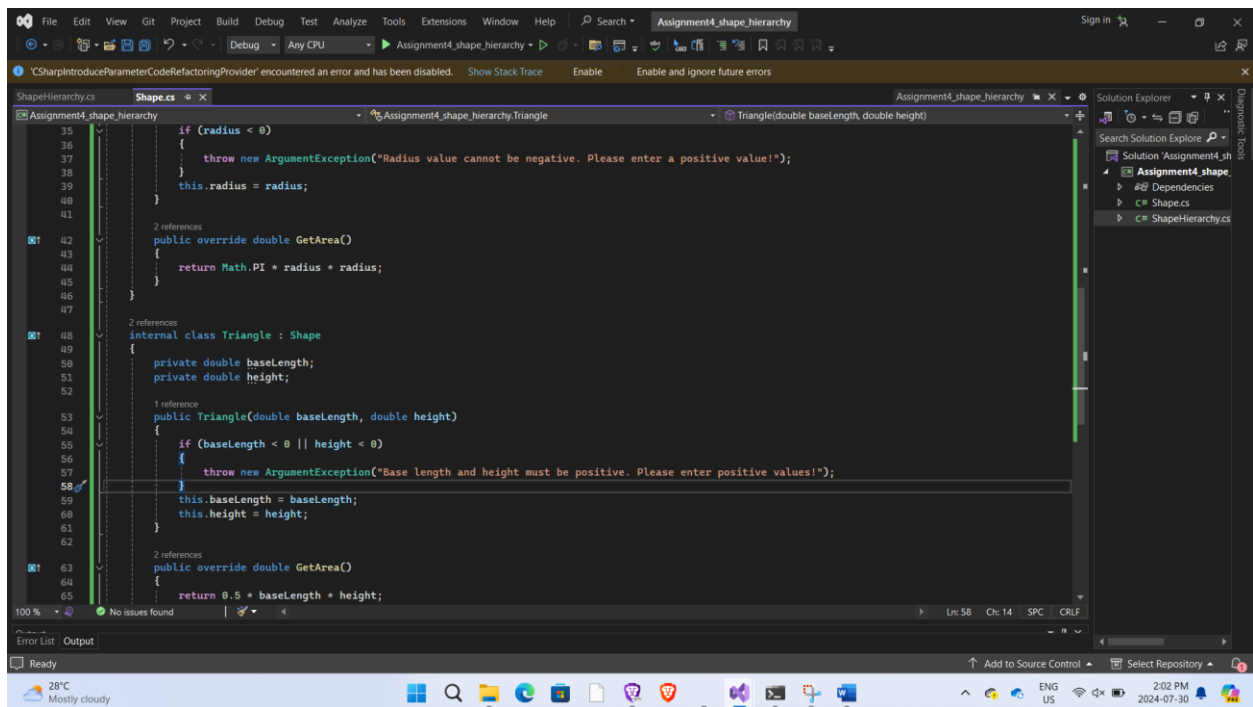
Student Id : C0930321

# Lab Question: Shape Hierarchy

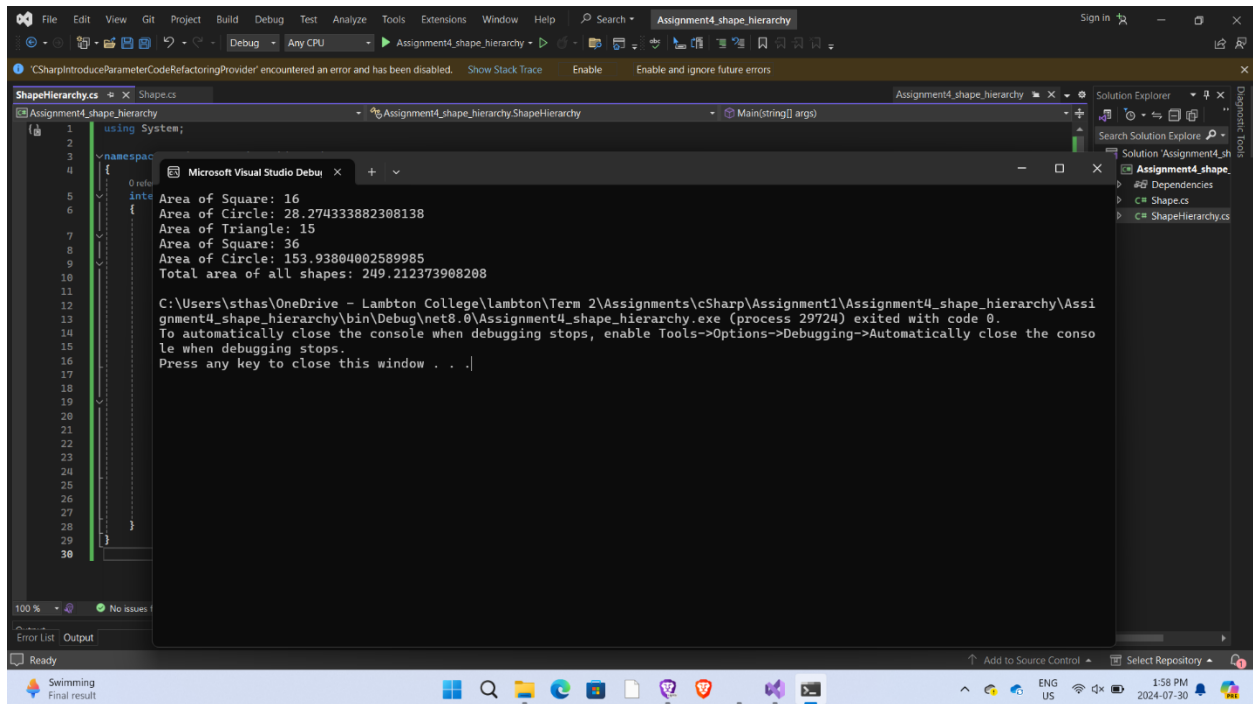
## # Code Screenshots



```
1 using System;
2
3 namespace Assignment4_shape_hierarchy
4 {
5     6 references
6     internal abstract class Shape
7     {
8         4 references
9         public abstract double GetArea();
10
11     3 references
12     internal class Square : Shape
13     {
14         private double side;
15
16         2 references
17         public Square(double side)
18         {
19             if (side < 0)
20             {
21                 throw new ArgumentException("The sides of the square cannot be negative. Please enter a positive value!");
22             }
23             this.side = side;
24         }
25
26         2 references
27         public override double GetArea()
28         {
29             return side * side;
30         }
31     }
32
33     3 references
34     internal class Circle : Shape
35     {
36     }
```



## # Output Screenshots



The screenshot shows the Microsoft Visual Studio IDE with a C# project named 'Assignment4\_shape\_hierarchy'. The 'Output' window displays the following text:

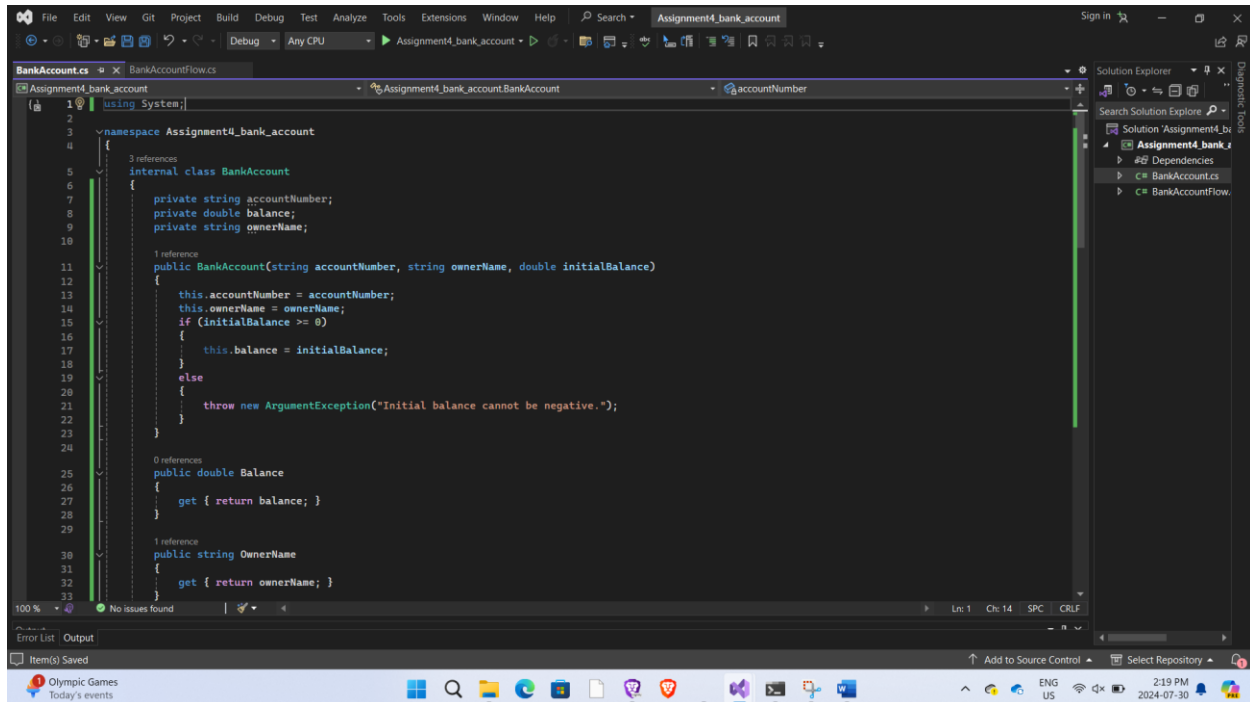
```
Area of Square: 16
Area of Circle: 28.274333882308138
Area of Triangle: 15
Area of Square: 36
Area of Circle: 153.93804002589985
Total area of all shapes: 249.212373968208

C:\Users\sthas\OneDrive - Lambton College\Lambton\Term 2\Assignments\cSharp\Assignment1\Assignment4_shape_hierarchy\Assignment4_shape_hierarchy\bin\Debug\net8.0\Assignment4_shape_hierarchy.exe (process 29724) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|
```

The 'Solution Explorer' on the right shows the project structure with files: 'Shape.cs' and 'ShapeHierarchy.cs'. The 'Error List' at the bottom left is empty, indicating no errors.

# Lab Question: Bank Account

## # Code screenshots



This screenshot shows the Visual Studio IDE with the `BankAccountFlow.cs` file open. The code defines a namespace `Assignment4_bank_account` containing an internal class `BankAccount`. The class has three private fields: `accountNumber` (string), `balance` (double), and `ownerName` (string). It includes a constructor `BankAccount(string accountNumber, string ownerName, double initialBalance)` that initializes the fields and checks if the initial balance is non-negative, throwing an `ArgumentException` if it is negative. Additionally, there are two public properties: `Balance` (double) and `OwnerName` (string), both with getter methods. The Solution Explorer on the right shows the project structure with `Assignment4_bank_account` as the main project, and `BankAccountFlow.cs` and `BankAccount.cs` as its files. The status bar at the bottom indicates 'No issues found'.

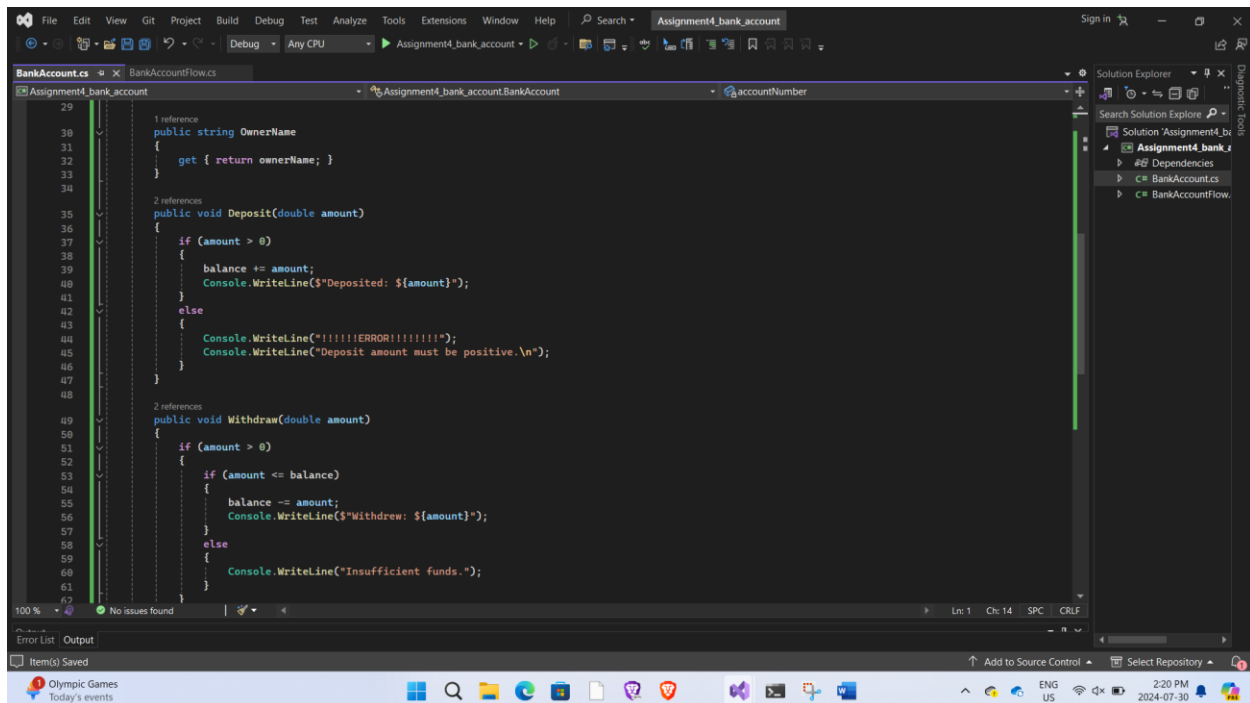
```
using System;

namespace Assignment4_bank_account
{
    internal class BankAccount
    {
        private string accountNumber;
        private double balance;
        private string ownerName;

        public BankAccount(string accountNumber, string ownerName, double initialBalance)
        {
            this.accountNumber = accountNumber;
            this.ownerName = ownerName;
            if (initialBalance >= 0)
            {
                this.balance = initialBalance;
            }
            else
            {
                throw new ArgumentException("Initial balance cannot be negative.");
            }
        }

        public double Balance
        {
            get { return balance; }
        }

        public string OwnerName
        {
            get { return ownerName; }
        }
    }
}
```

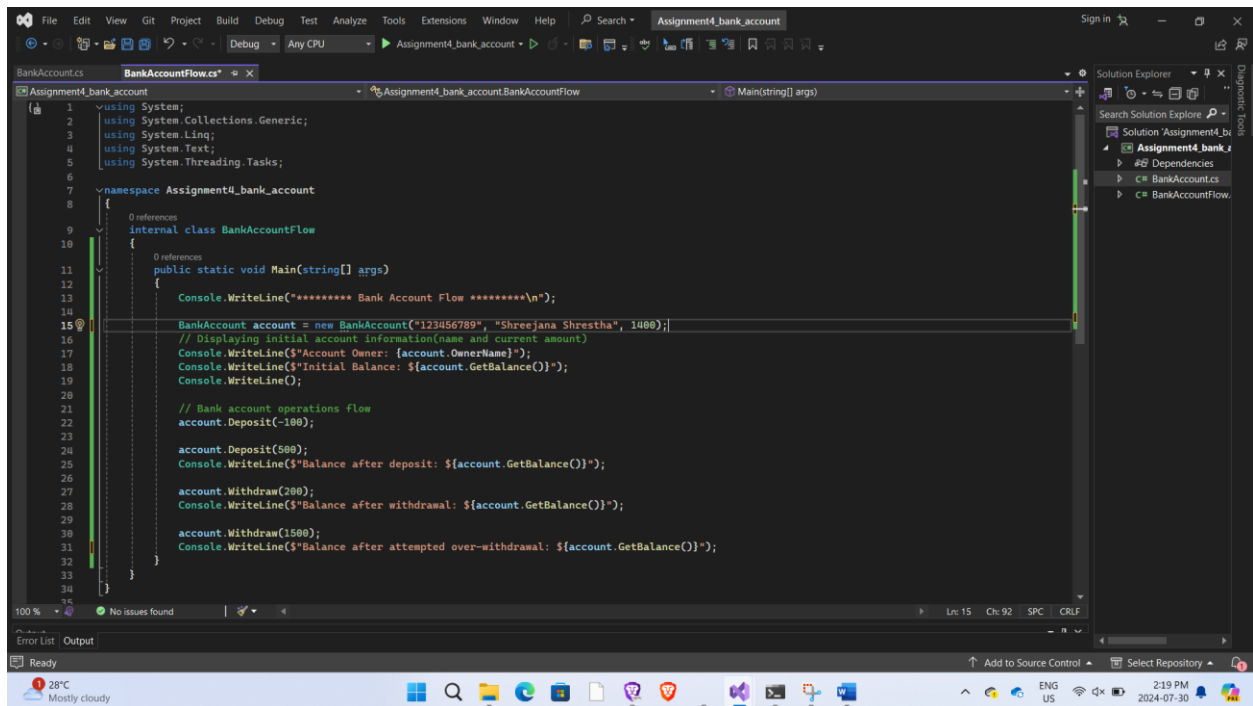
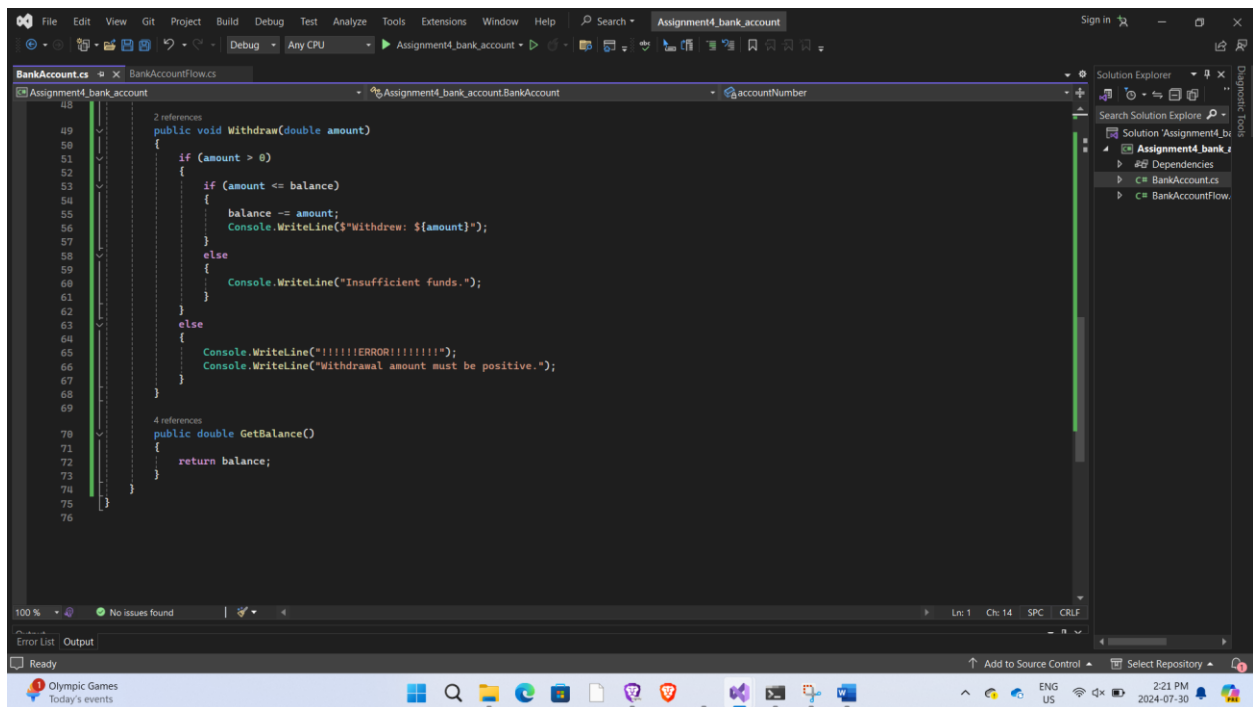


This screenshot shows the Visual Studio IDE with the `BankAccountFlow.cs` file open, displaying the `Deposit` and `Withdraw` methods of the `BankAccount` class. The `Deposit` method takes a `double amount` as input. It checks if the amount is greater than 0. If so, it adds the amount to the `balance` and prints a message to the console: `Console.WriteLine($"Deposited: ${amount}");`. If the amount is not greater than 0, it prints an error message: `Console.WriteLine("!!!!!!ERROR!!!!!!");` and a warning: `Console.WriteLine("Deposit amount must be positive.\n");`. The `Withdraw` method also takes a `double amount` as input. It checks if the amount is greater than 0. If so, it checks if the amount is less than or equal to the current `balance`. If yes, it subtracts the amount from the `balance` and prints a message: `Console.WriteLine($"Withdraw: ${amount}");`. If not, it prints an error message: `Console.WriteLine("Insufficient funds.");`. The Solution Explorer on the right shows the project structure. The status bar at the bottom indicates 'No issues found'.

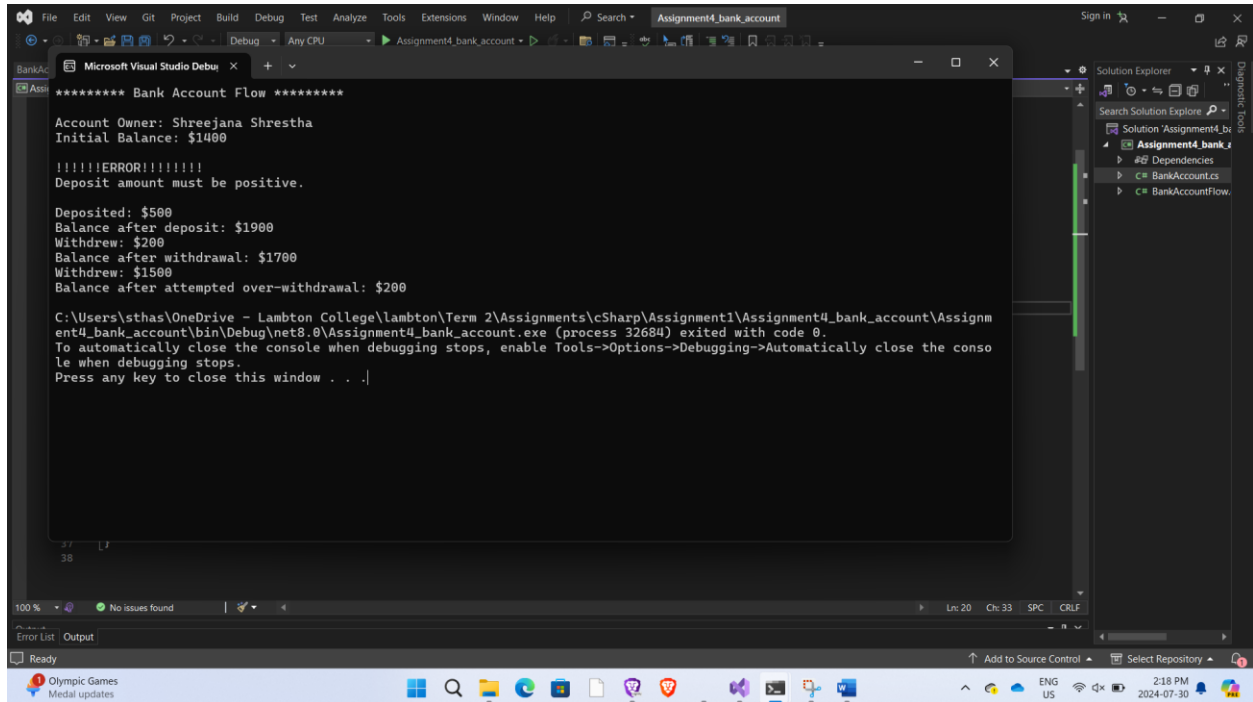
```
public string OwnerName
{
    get { return ownerName; }
}

public void Deposit(double amount)
{
    if (amount > 0)
    {
        balance += amount;
        Console.WriteLine($"Deposited: ${amount}");
    }
    else
    {
        Console.WriteLine("!!!!!!ERROR!!!!!!");
        Console.WriteLine("Deposit amount must be positive.\n");
    }
}

public void Withdraw(double amount)
{
    if (amount > 0)
    {
        if (amount <= balance)
        {
            balance -= amount;
            Console.WriteLine($"Withdraw: ${amount}");
        }
        else
        {
            Console.WriteLine("Insufficient funds.");
        }
    }
}
```



## # Output Screenshots



```
***** Bank Account Flow *****  
  
Account Owner: Shreejana Shrestha  
Initial Balance: $1400  
  
!!!!!!ERROR!!!!!!  
Deposit amount must be positive.  
  
Deposited: $500  
Balance after deposit: $1900  
Withdrew: $200  
Balance after withdrawal: $1700  
Withdrew: $1500  
Balance after attempted over-withdrawal: $200  
  
C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\cSharp\Assignment1\Assignment4_bank_account\Assignm  
ent4_bank_account\bin\Debug\net8.0\Assignment4_bank_account.exe (process 32684) exited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso  
le when debugging stops.  
Press any key to close this window . . .]
```