NORTH SOUTH UNIVERSITY

Department of Electrical & Computer Engineering

Assignment On

Course Code: *CSE425*
Course Title: *Concepts of Programming Language*

Submitted by_

Name	: Sajan Kumer Sarker
ID#	: 2111131642
Email	: sajan.sarker@northsouth.edu
Section	: 08
Assignment Title	: Homework 2
Submission Date	: 23rd, May 2024

Submitted to_

Instructor	: Dr. Mohammad Abdur Rouf (MDAR)
Email	: mohammad.rouf@northsouth.edu

1. Write a Program to Find the Length of the string without using strlen() Function.

Solution: The Program to Find the Length of the string without using strlen() Function is given below using C++ programming Language:

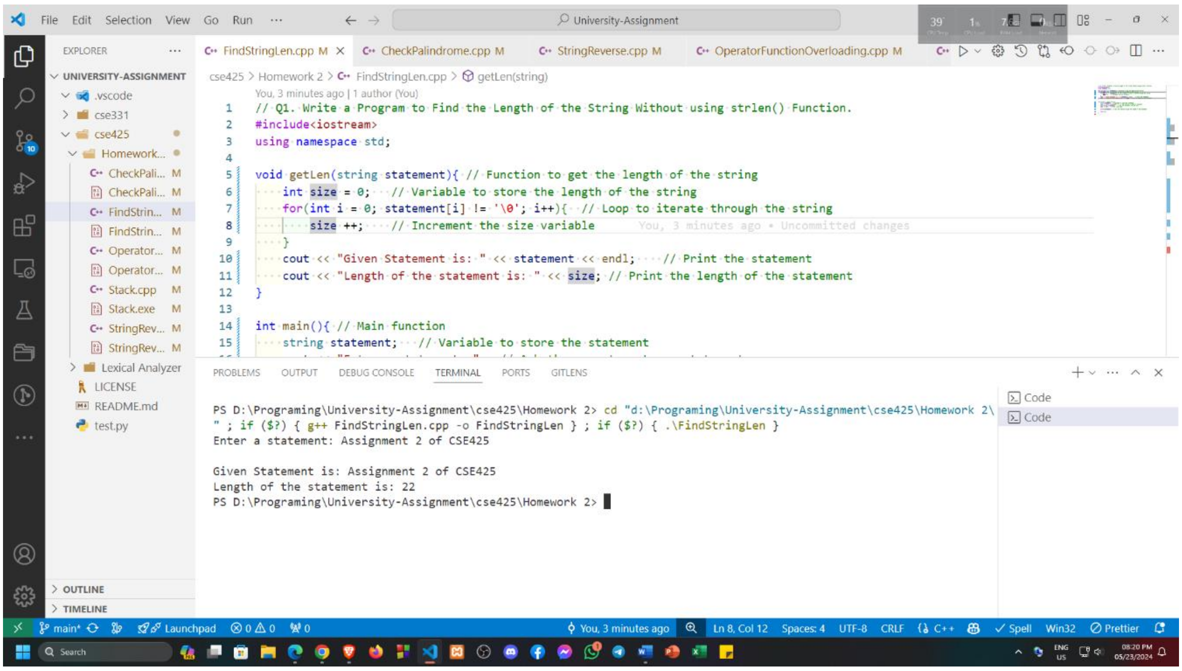
Source Code:

```cpp
// File name: FindStringLen.cpp
// Q1. Write a Program to Find the Length of the String Without using strlen() Function.
#include<iostream>
using namespace std;

void getLen(string statement){ // Function to get the length of the string
    int size = 0;    // Variable to store the length of the string
    for(int i = 0; statement[i] != '\0'; i++){   // Loop to iterate through the string
        size ++;     // Increment the size variable
    }
    cout << "Given Statement is: " << statement << endl;     // Print the statement
    cout << "Length of the statement is: " << size; // Print the length of the statement
}

int main(){ // Main function
    string statement;    // Variable to store the statement
    cout << "Enter a statement: ";   // Ask the user to enter a statement
    getline(cin, statement);     // Get the statement from the user
    cout << endl;    // Print a new line

    getLen(statement);   // Call the function to get the length of the statement

    return 0;
}
```

Input & Output of the Code:

```
PS D:\Programing\University-Assignment> cd "d:\Programing\University-
Assignment\cse425\Homework 2\" ; if ($?) { g++ FindStringLen.cpp -o FindStringLen }
; if ($?) { .\FindStringLen }
Enter a statement: Assignment 2 of CSE425

Given Statement is: Assignment 2 of CSE425
Length of the statement is: 22
```

Screenshot of The Output:



```cpp
// Q1. Write a Program to Find the Length of the String Without using strlen() Function.
#include<iostream>
using namespace std;

void getLen(string statement){ // Function to get the length of the string
    int size = 0;    // Variable to store the length of the string
    for(int i = 0; statement[i] != '\0'; i++){  // Loop to iterate through the string
        size ++;    // Increment the size variable
    }
    cout << "Given Statement is: " << statement << endl;    // Print the statement
    cout << "Length of the statement is: " << size; // Print the length of the statement
}

int main(){ // Main function
    string statement;   // Variable to store the statement
```

```
PS D:\Programing\University-Assignment\cse425\Homework 2> cd "d:\Programing\University-Assignment\cse425\Homework 2\
" ; if ($?) { g++ FindStringLen.cpp -o FindStringLen } ; if ($?) { .\FindStringLen }
Enter a statement: Assignment 2 of CSE425

Given Statement is: Assignment 2 of CSE425
Length of the statement is: 22
PS D:\Programing\University-Assignment\cse425\Homework 2>
```

2. Write a Program to Check Palindrome.

Solution: The program to Check Palindrome is given below using C++ Programming Language:

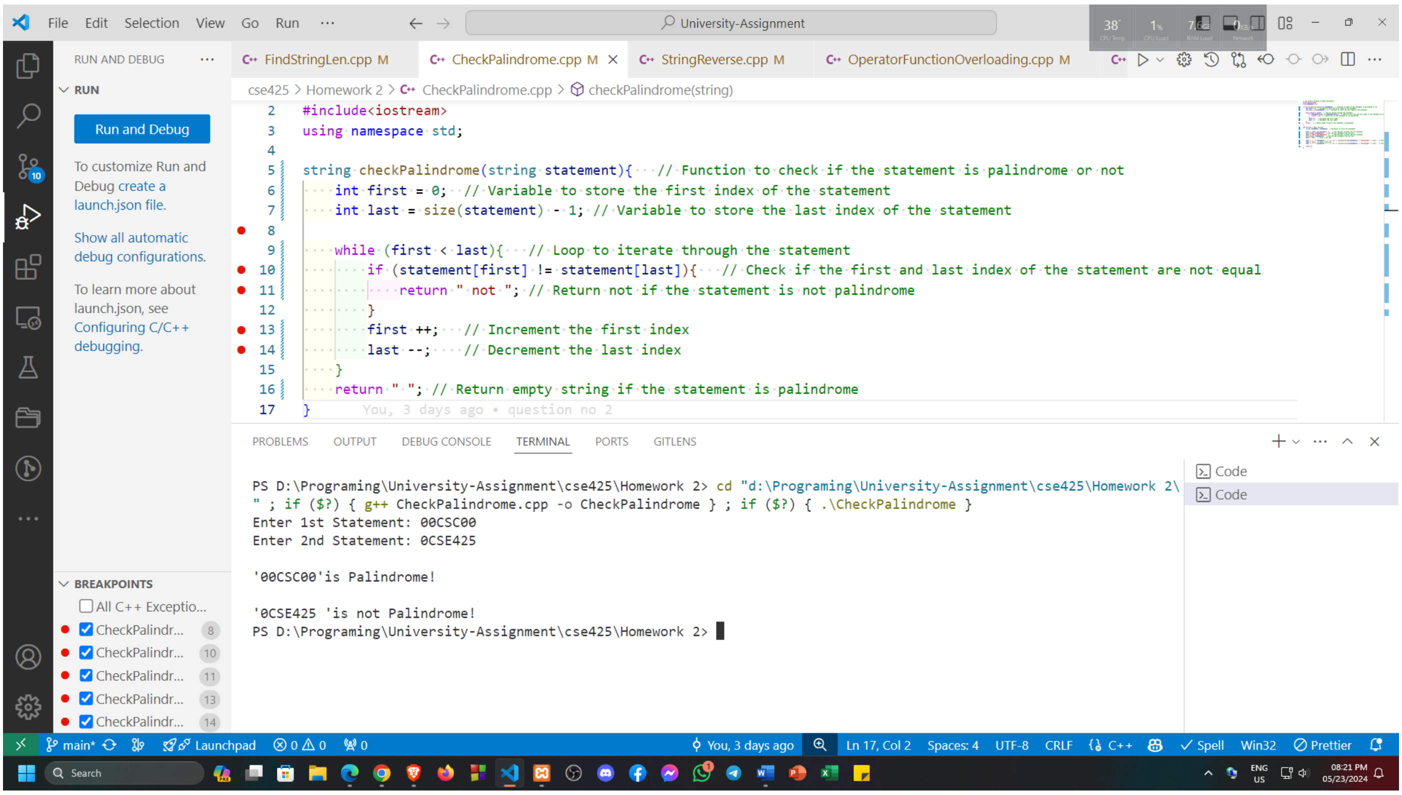
Source Code:

```cpp
// File name: CheckPalindrome.cpp
// Q2. Write a Program to Check Palindrome.
#include<iostream>
using namespace std;

string checkPalindrome(string statement){   // Function to check if the statement is palindrome or not
    int first = 0;  // Variable to store the first index of the statement
    int last = size(statement) - 1; // Variable to store the last index of the statement

    while (first < last){   // Loop to iterate through the statement
        if (statement[first] != statement[last]){   // Check if the first and last index of the statement are not equal
            return " not "; // Return not if the statement is not palindrome
        }
        first ++;   // Increment the first index
        last --;    // Decrement the last index
    }
    return " "; // Return empty string if the statement is palindrome
}

int main(){ // Main function
    string statement1, statement2;  // Variables to store the statements

    cout << "Enter 1st Statement: ";    // Ask the user to enter the 1st statement
    getline(cin, statement1);   // Get the 1st statement from the user
    cout << "Enter 2nd Statement: ";    // Ask the user to enter the 2nd statement
    getline(cin, statement2);   // Get the 2nd statement from the user
    cout << endl;   // Print a new line

    cout << "'" << statement1 << "'" << "is" << checkPalindrome(statement1) << "Palindrome!" << endl;   // Print the result of the 1st statement
    cout << endl;   // Print a new line
    cout << "'" << statement2 << "'" << "is" << checkPalindrome(statement2) << "Palindrome!" << endl;   // Print the result of the 2nd statement

    return 0;
}
```

Input & Output of the Code:

```
PS D:\Programing\University-Assignment\cse425\Homework 2> cd "d:\Programing\University-Assignment\cse425\Homework 2\" ; if ($?) { g++ CheckPalindrome.cpp -o CheckPalindrome } ; if ($?) { .\CheckPalindrome }

Enter 1st Statement: 00CSC00
Enter 2nd Statement: 0CSE425

'00CSC00' is Palindrome!

'0CSE425' is not Palindrome!
```

Screenshot of the Output:



File Edit Selection View Go Run
University-Assignment
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Show all automatic debug configurations.
To learn more about launch.json, see Configuring C/C++ debugging.
BREAKPOINTS
All C++ Exceptio...
CheckPalindr... 8
CheckPalindr... 10
CheckPalindr... 11
CheckPalindr... 13
CheckPalindr... 14
FindStringLen.cpp M
CheckPalindrome.cpp M
StringReverse.cpp M
OperatorFunctionOverloading.cpp M
cse425 > Homework 2 > CheckPalindrome.cpp > checkPalindrome(string)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
Code
Code
main* Launchpad 0 0 0
You, 3 days ago
Ln 17, Col 2 Spaces: 4 UTF-8 CRLF C++ Spell Win32 Prettier
Search
ENG US
08:21 PM 05/23/2024

```cpp
#include<iostream>
using namespace std;

string checkPalindrome(string statement){   // Function to check if the statement is palindrome or not
    int first = 0;  // Variable to store the first index of the statement
    int last = size(statement) - 1; // Variable to store the last index of the statement

    while (first < last){   // Loop to iterate through the statement
        if (statement[first] != statement[last]){   // Check if the first and last index of the statement are not equal
            return " not "; // Return not if the statement is not palindrome
        }
        first ++;   // Increment the first index
        last --;    // Decrement the last index
    }
    return " "; // Return empty string if the statement is palindrome
}       You, 3 days ago • question no 2
```

```
PS D:\Programing\University-Assignment\cse425\Homework 2> cd "d:\Programing\University-Assignment\cse425\Homework 2\
" ; if ($?) { g++ CheckPalindrome.cpp -o CheckPalindrome } ; if ($?) { .\CheckPalindrome }
Enter 1st Statement: 00CSC00
Enter 2nd Statement: 0CSE425

'00CSC00'is Palindrome!

'0CSE425 'is not Palindrome!
PS D:\Programing\University-Assignment\cse425\Homework 2> 
```

3. Write a C++ Program to Print the Given String in Reverse Order Using Recursion.

Solution: The program to Print the Given String in Reverse Order Using Recursion is given below using C++ Programming Language:

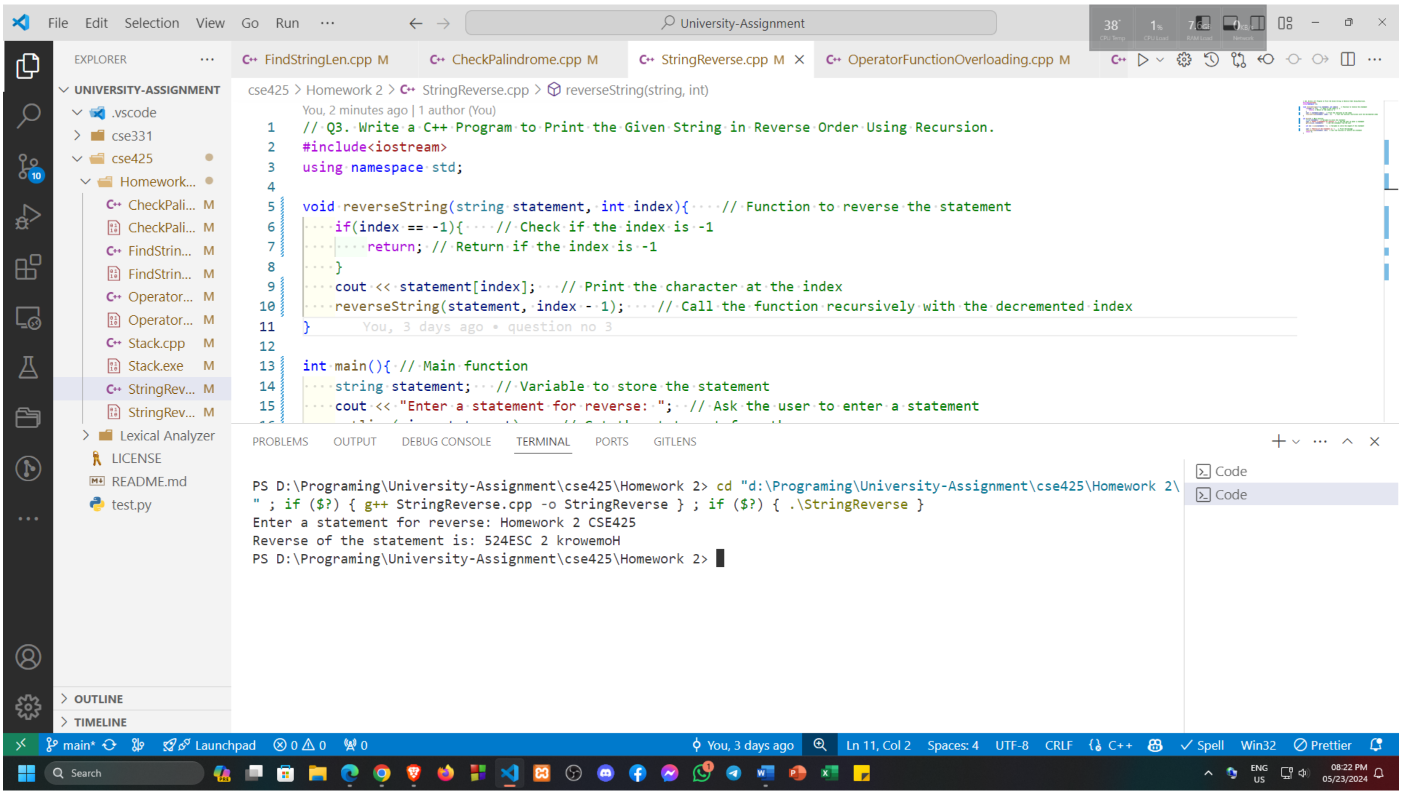
Source Code:

```
// File name: StringReverse.cpp
// Q3. Write a C++ Program to Print the Given String in Reverse Order Using Recursion.
#include<iostream>
using namespace std;

void reverseString(string statement, int index){    // Function to reverse the statement
    if(index == -1){    // Check if the index is -1
        return; // Return if the index is -1
    }
    cout << statement[index];   // Print the character at the index
    reverseString(statement, index - 1);    // Call the function recursively with the decremented index
}

int main(){ // Main function
    string statement;   // Variable to store the statement
    cout << "Enter a statement for reverse: ";  // Ask the user to enter a statement
    getline(cin, statement);    // Get the statement from the user

    int len = size(statement) - 1;  // Variable to store the length of the statement

    cout << "Reverse of the statement is: ";    // Print the message
    reverseString(statement, len);  // Call the function to reverse the statement
    return 0;
}
```

Input & Output of the Code:

```
PS D:\Programing\University-Assignment\cse425\Homework 2> cd
"d:\Programing\University-Assignment\cse425\Homework 2\" ; if ($?) { g++
StringReverse.cpp -o StringReverse } ; if ($?) { .\StringReverse }

Enter a statement for reverse: Homework 2 CSE425
Reverse of the statement is: 524ESC 2 krowemoH
```

Screenshot of the Output:



University-Assignment
FindStringLen.cpp M
CheckPalindrome.cpp M
StringReverse.cpp M
OperatorFunctionOverloading.cpp M
cse425 > Homework 2 > StringReverse.cpp > reverseString(string, int)
You, 2 minutes ago | 1 author (You)
You, 3 days ago • question no 3
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS

```cpp
// Q3. Write a C++ Program to Print the Given String in Reverse Order Using Recursion.
#include<iostream>
using namespace std;

void reverseString(string statement, int index){    // Function to reverse the statement
    if(index == -1){    // Check if the index is -1
        return; // Return if the index is -1
    }
    cout << statement[index];   // Print the character at the index
    reverseString(statement, index - 1);    // Call the function recursively with the decremented index
}

int main(){ // Main function
    string statement;   // Variable to store the statement
    cout << "Enter a statement for reverse: ";  // Ask the user to enter a statement
```

```
PS D:\Programing\University-Assignment\cse425\Homework 2> cd "d:\Programing\University-Assignment\cse425\Homework 2\
" ; if ($?) { g++ StringReverse.cpp -o StringReverse } ; if ($?) { .\StringReverse }
Enter a statement for reverse: Homework 2 CSE425
Reverse of the statement is: 524ESC 2 krowemoH
PS D:\Programing\University-Assignment\cse425\Homework 2>
```

4. Write a Program to Implement the Concept of Operator and Function Overloading.

Solution: The program to Implement the Concept of Operator and Function Overloading is given below using C++ Programming Language:

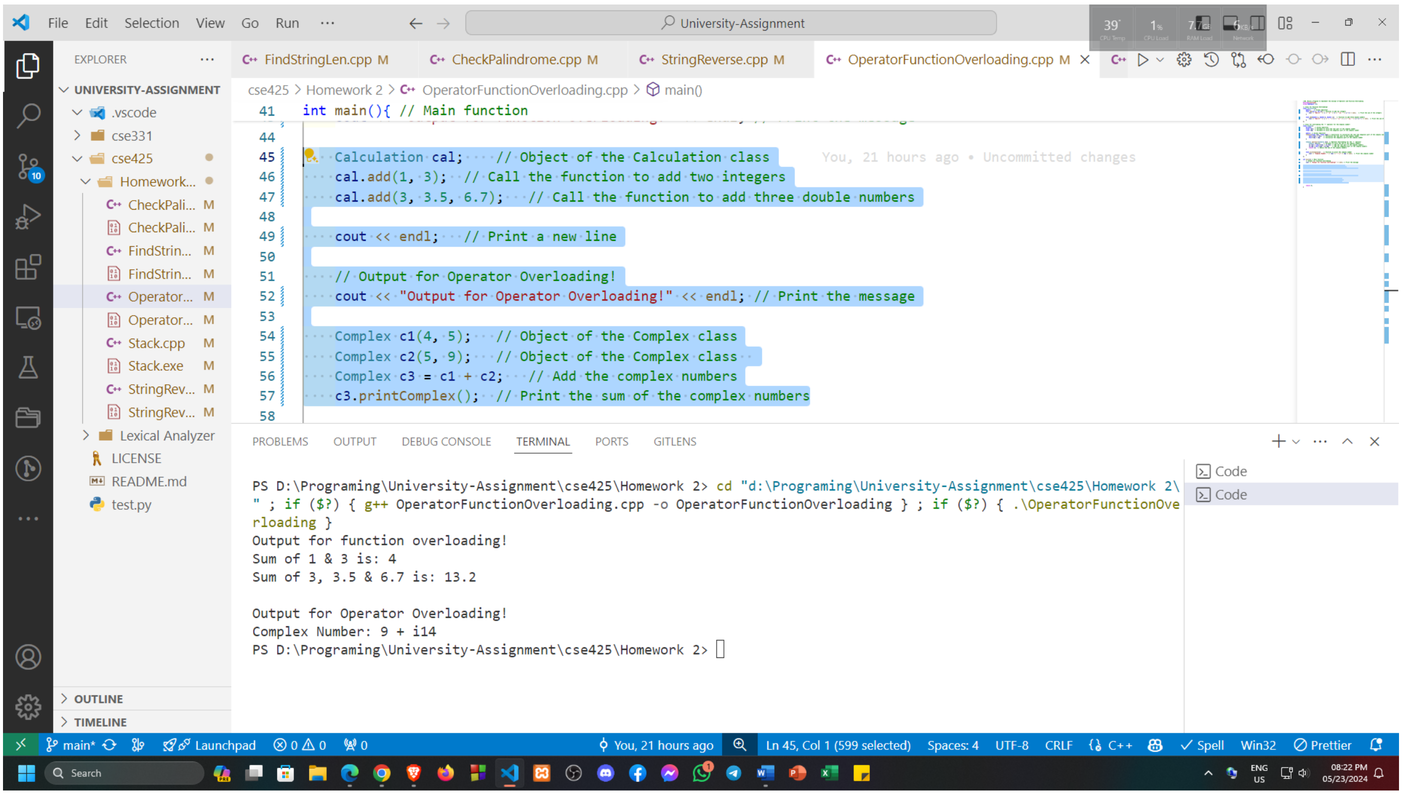
Source Code:

```cpp
// File name: OperatorFunctionOverloading.cpp
// Q4. Write a Program to Implement the Concept of Operator and Function Overloading
#include<iostream>
using namespace std;

// Class for Function Overloading!
class Calculation{
    public:    // Access Specifier
    void add(int a, int b){ // Function to add two integers
        cout << "Sum of " << a << " & " << b << " is: " << a + b << endl;    // Print the sum of the integers
    }

    void add(double a, double b, double c){    // Function to add three double numbers
        cout << "Sum of " << a << ", " << b << " & " << c << " is: " << a + b + c << endl;  // Print the sum of the double numbers
    }
};

// Class for overloading the '+' operator for the Complex number!
class Complex{
    private:    // Access Specifier
    float real; // Variable to store the real part of the complex number
    float img;  // Variable to store the imaginary part of the complex number

    public: // Access Specifier
    Complex(float real, float img){ // Constructor to initialize the real and imaginary part of the complex number
        this->real = real;  // Initialize the real part of the complex number
        this->img = img;    // Initialize the imaginary part of the complex number
    }

    Complex operator+(Complex c2){  // Operator Overloading for the '+' operator
        Complex c3(0,0);    // Variable to store the sum of the complex numbers
        c3.real = this->real + c2.real; // Add the real part of the complex numbers
        c3.img = this->img + c2.img;    // Add the imaginary part of the complex numbers
        return c3;  // Return the sum of the complex numbers
    }

    void printComplex(){    // Function to print the complex number
        cout << “Complex Number: ” real << " + i" << img << endl;  // Print the complex number
    }
};

int main(){ // Main function
    // Output for function overloading
    cout << "Output for function overloading!" << endl; // Print the message

    Calculation cal;    // Object of the Calculation class
    cal.add( 1, 3);  // Call the function to add two integers
    cal.add( 3, 3.5, 6.7);    // Call the function to add three double numbers

    cout << endl;    // Print a new line

    // Output for Operator Overloading!
    cout << "Output for Operator Overloading!" << endl; // Print the message

    Complex c1( 4, 5);    // Object of the Complex class
    Complex c2( 5, 9);    // Object of the Complex class
    Complex c3 = c1 + c2;    // Add the complex numbers
    c3.printComplex();  // Print the sum of the complex numbers

    return 0;
}
```

Output of the Code:

```
PS D:\Programing\University-Assignment\cse425\Homework 2> cd
"d:\Programing\University-Assignment\cse425\Homework 2\" ; if ($?) { g++
OperatorFunctionOverloading.cpp -o OperatorFunctionOverloading } ; if ($?) {
.\OperatorFunctionOverloading }

Output for function overloading!
Sum of 1 & 3 is: 4
Sum of 3, 3.5 & 6.7 is: 13.2

Output for Operator Overloading!
Complex Number: 9 + i14
```

Screenshot of the Output:



```
int main(){ // Main function

    Calculation cal;    // Object of the Calculation class
    cal.add(1, 3);  // Call the function to add two integers
    cal.add(3, 3.5, 6.7);   // Call the function to add three double numbers

    cout << endl;   // Print a new line

    // Output for Operator Overloading!
    cout << "Output for Operator Overloading!" << endl; // Print the message

    Complex c1(4, 5);   // Object of the Complex class
    Complex c2(5, 9);   // Object of the Complex class
    Complex c3 = c1 + c2;   // Add the complex numbers
    c3.printComplex();  // Print the sum of the complex numbers

```

```
PS D:\Programing\University-Assignment\cse425\Homework 2> cd "d:\Programing\University-Assignment\cse425\Homework 2\
" ; if ($?) { g++ OperatorFunctionOverloading.cpp -o OperatorFunctionOverloading } ; if ($?) { .\OperatorFunctionOve
rloading }
Output for function overloading!
Sum of 1 & 3 is: 4
Sum of 3, 3.5 & 6.7 is: 13.2

Output for Operator Overloading!
Complex Number: 9 + i14
PS D:\Programing\University-Assignment\cse425\Homework 2> 
```

5. Write a Program for Implementation of Stacks Using an Array using Object Oriented Modeling.

Solution: The program to Implementation of Stacks Using an Array using Object Oriented Modeling is given below using C++ Programming Language:

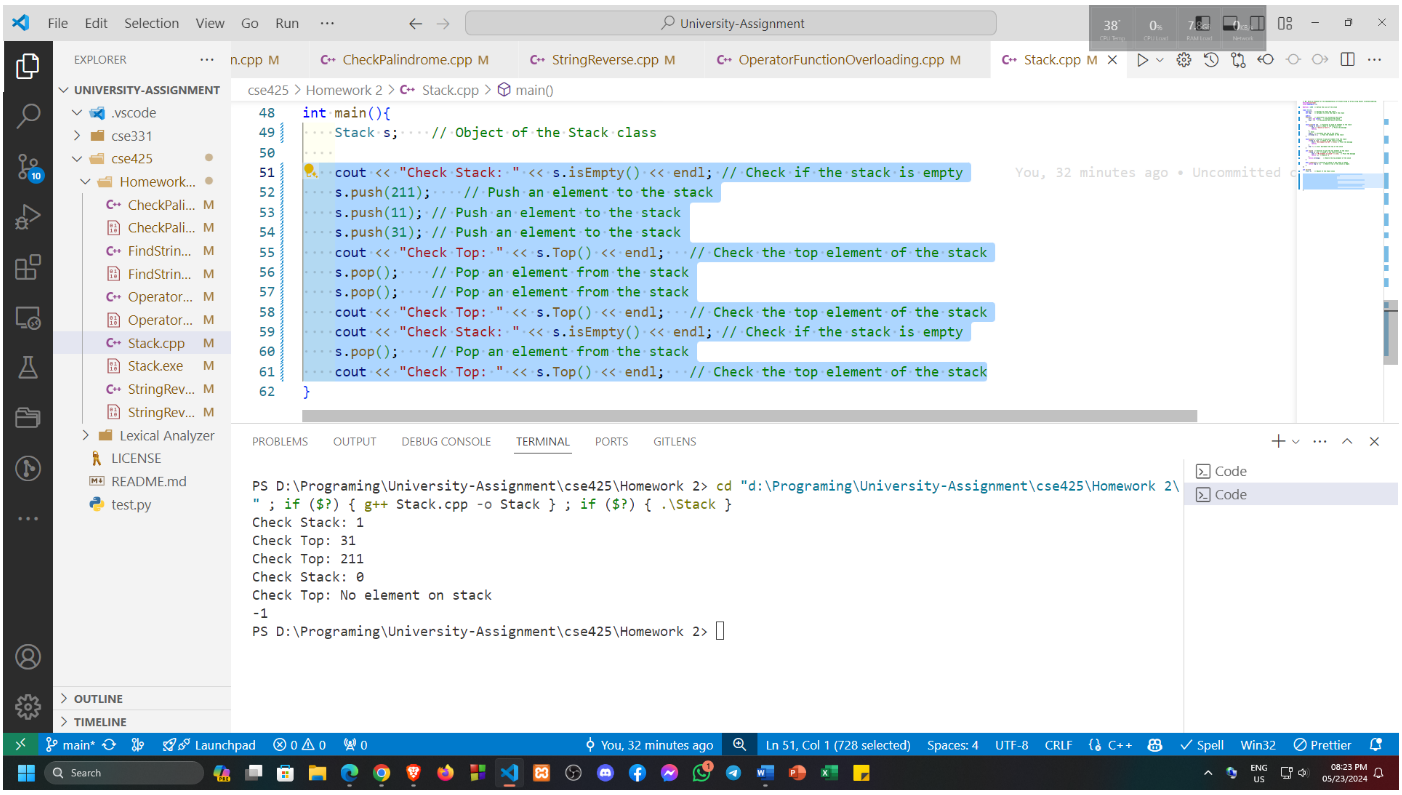
Source Code:

```cpp
// File name: Stack.cpp
// Q5. Write a Program for the Implementation of Stacks Using an Array using object oriented modeling.
#include<iostream>
using namespace std;

#define n 1000  // Define the size of the stack

class Stack{
    int* arr;   // Pointer to store the stack
    int top;    // Variable to store the top of the stack

    public:
    Stack(){    // Constructor to initialize the stack
        arr = new int[n];   // Initialize the stack
        top = -1;   // Initialize the top of the stack
    }

    void push(int x){   // Function to push an element to the stack
        if(top == n-1){ // Check if the stack is full
            cout << "Stack is Full!";   // Print the message
            return;
        }
        // else:
        top++;  // Increment the top of the stack
        arr[top] = x;   // Push the element to the stack
    }

    void pop(){ // Function to pop an element from the stack
        if(top == -1){  // Check if the stack is empty
            cout<< "No Element to POP" << endl; // Print the message
            return;
        }
        top --; // else: decrement the top of the stack
    }

    int Top(){  // Function to get the top element of the stack
        if(top == -1){  // Check if the stack is empty
            cout << "No element on stack" << endl;  // Print the message
            return -1;  // Return -1
        }
        return arr[top];    // Return the top element of the stack
    }

    bool isEmpty(){ // Function to check if the stack is empty
        return top == -1;   // Return true if the stack is empty
    }
};

int main(){
    Stack s;    // Object of the Stack class

    cout << "Check Stack: " << s.isEmpty() << endl; // Check if the stack is empty
    s.push(211);    // Push an element to the stack
    s.push(11); // Push an element to the stack
    s.push(31); // Push an element to the stack
    cout << "Check Top: " << s.Top() << endl;    // Check the top element of the stack
    s.pop();     // Pop an element from the stack
    s.pop();     // Pop an element from the stack
    cout << "Check Top: " << s.Top() << endl;    // Check the top element of the stack
    cout << "Check Stack: " << s.isEmpty() << endl; // Check if the stack is empty
    s.pop();     // Pop an element from the stack
    cout << "Check Top: " << s.Top() << endl;    // Check the top element of the stack
}
```

Output of the Code:

```
PS D:\Programing\University-Assignment\cse425\Homework 2> cd
"d:\Programing\University-Assignment\cse425\Homework 2\" ; if ($?) { g++ Stack.cpp -o
Stack } ; if ($?) { .\Stack }

Check Stack: 1
Check Top: 31
Check Top: 211
Check Stack: 0
Check Top: No element on stack
-1
```

Screenshot of the Output:



```
int main(){
    Stack s;    // Object of the Stack class

    cout << "Check Stack: " << s.isEmpty() << endl; // Check if the stack is empty
    s.push(211);    // Push an element to the stack
    s.push(11); // Push an element to the stack
    s.push(31); // Push an element to the stack
    cout << "Check Top: " << s.Top() << endl;   // Check the top element of the stack
    s.pop();    // Pop an element from the stack
    s.pop();    // Pop an element from the stack
    cout << "Check Top: " << s.Top() << endl;   // Check the top element of the stack
    cout << "Check Stack: " << s.isEmpty() << endl; // Check if the stack is empty
    s.pop();    // Pop an element from the stack
    cout << "Check Top: " << s.Top() << endl;   // Check the top element of the stack
}
```

```
PS D:\Programing\University-Assignment\cse425\Homework 2> cd "d:\Programing\University-Assignment\cse425\Homework 2\
" ; if ($?) { g++ Stack.cpp -o Stack } ; if ($?) { .\Stack }
Check Stack: 1
Check Top: 31
Check Top: 211
Check Stack: 0
Check Top: No element on stack
-1
PS D:\Programing\University-Assignment\cse425\Homework 2> 
```