SmallSEOTools

Plagiarism Scan Report By SmallSEOTools

Report Generated on: Oct 29,2024

0%

Plagiarized Content

0%

Exact Plagiarized

0%

Partial Plagiarized

100%

Unique Content

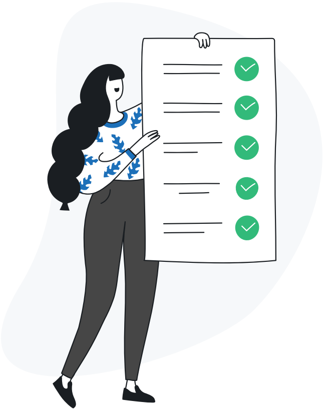
Total Words: 336 **Total Characters:** 2229 **Plagiarized Sentences:** 0 **Unique Sentences:** 5 (100%)

Content Checked for Plagiarism

```
#include
using namespace std;
// Node class
class Node
{
private:
int data;

public:
Node *next;
Node(int val) : data(val), next(nullptr) {} // Parametrised constructor for Creating a node
int getData();
};

int Node::getData()
{
return data; // returns data
}

class Stack
{
private:
Node *top;

public:
Stack();
void push(int);
int pop();
int peek();
bool isEmpty();
~Stack();
};
// constructor
Stack::Stack()
{
top = nullptr; // If top = nullptr, then the stack is empty
```

```
}

void Stack::push(int x)
{
Node *t = new Node(x); // creates a node in heap memory
t->next = top; // Linking the new node
top = t; // pointing top to the new node
cout << "Push Operation Performed!" << endl;
}

int Stack::pop()
{
if (isEmpty())
{
cout << "Stack Underflow!" << endl;
return -1;
}
else
{
Node *t = top;
int x = t->getData();
top = top->next;
delete t; // deallocate memory of the top node as the element is popped
cout << "Pop Operation Performed!" << endl;
return x;
}
}

int Stack::peek()
{
if (!isEmpty())
return top->getData();
else
return -1;
}

bool Stack::isEmpty()
{
return top == nullptr;
}

Stack::~Stack()
{
while (!isEmpty())
{
pop();
}
}

class Demo
{
public:
int factorial(int, Stack &);
};
int Demo::factorial(int n, Stack &callStack)
{
callStack.push(n); // Push current function call onto stack
cout << "Top element in the Stack : " << callStack.peek() << endl;
```

```
if (n == 0 || n == 1)
{
callStack.pop();
cout << "Top element in the Stack : " << callStack.peek() << endl;
return 1;
}

// Recursive case
int result = n * factorial(n - 1, callStack);

callStack.pop();
cout << "Top element in the Stack : " << callStack.peek() << endl;

return result;
}

int main()
{
int num;
Demo obj;
cout << "Enter a number to compute its factorial: ";
cin >> num;

Stack callStack;

int result = obj.factorial(num, callStack);
cout << "Factorial of " << num << " is " << result << endl;

return 0;
}
```



No Plagiarism Found