Plagiarism Scan Report By SmallSEOTools

Report Generated on: Oct 29,2024

Plagiarized Content

0% Exact Plagiarized

0% Partial Plagiarized

Unique Content

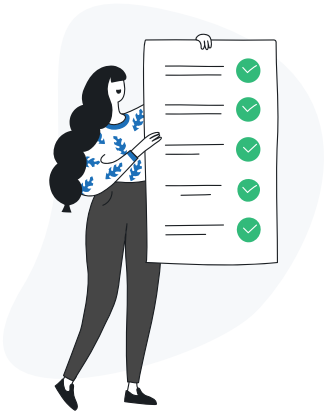
Total Words: 412 **Total Characters:** 2623 **Plagiarized Sentences:** 0 **Unique Sentences:** 4 (100%)

Content Checked for Plagiarism

```
#include
using namespace std;

class stack
{
private:
int size; // stores total no. of elements in stack
int top; // points to the top of the stack
int *s; // pointer to declare array in heap memory

public:
stack(int);
void push(int);
int pop();
int peek();
int isEmpty();
int isFull();
~stack();
};
// paremetrized constructor
stack::stack(int x)
{
size = x;
s = new int[size]; // creates an array in heap memory of size passed in constructor as param.
top = -1; // if top = -1, then the stack is empty
}
// push operation in stack
void stack::push(int x)
{
if (isFull())
{
cout << "Stack Full!" << endl;
return;
}
else
{
```

```
top++; // increment top and let it point to top of the array
s[top] = x; // copy the element to the top
cout << "push operation performed" << endl;
}
}
// removes the top-most element from the stack
int stack::pop()
{
int temp = -1;
if (isEmpty())
{
temp = -1;
}
else
{
temp = s[top]; // copying the element in temp variable before removing to return it
top--; // decrements top so that the top element is popped
cout << "pop operation performed" << endl;
}
return temp;
}
// returns the top element from the stack
int stack::peek()
{
if (isEmpty())
{
return -1;
}
else
{
return s[top];
}
}

// Checks whether is the stack is empty or not
int stack::isEmpty()
{
if (top == -1)
{
return 1;
}
else
{
return 0;
}
}
// Checks whether is the stack is full or not
int stack::isFull()
{
if (top == size - 1)
{
return 1;
}
else
{
return 0;
}
}
```

```
stack::~stack()
{
delete[] s;
}

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 : "<> num;

stack callStack(100);

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

return 0;
}
```



No Plagiarism Found