DAILY ONLINE ACTIVITIES SUMMARY

Date:	**23-06-2020**	**Name:**	**Rakesh M Kotian**
Sem & Sec	**8 th sec-b**	**USN:**	**4al16cs072**

Online Test Summary

Subject			
Max. Marks		**Score**	

Certification Course Summary

Course	**Python for machine learning**		
Certificate Provider	**Great learning**	**Duration**	**6 hours**

Coding Challenges

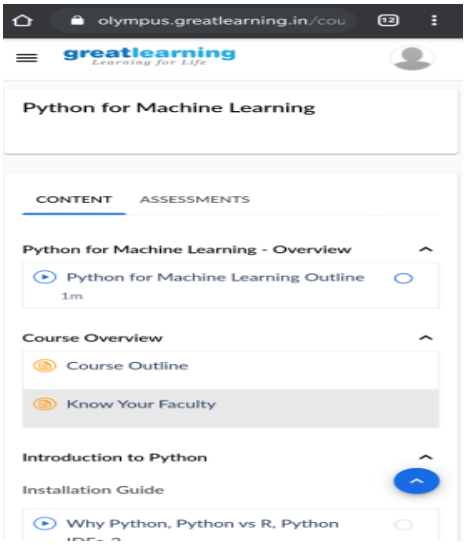
Problem Statement:
Postfix to Infix

Status:solved

Uploaded the report in Github	**yes**
If yes Repository name	**Rakeshkotian08**
Uploaded the report in slack	**yes**

Online Test Details: (Attach the snapshot and briefly write the report for the same)

Certification Course Details: (Attach the snapshot and briefly write the report for the same)




```
// CPP Program to convert postfix to prefix
#include <bits/stdc++.h>
using namespace std;

// function to check if character is operator or not
bool isOperator(char x)
{
    switch (x) {
    case '+':
    case '-':
```

```
    case '/':
    case '*':
        return true;
    }
    return false;
}

// Convert postfix to Prefix expression
string postToPre(string post_exp)
{
    stack<string> s;

    // length of expression
    int length = post_exp.size();

    // reading from right to left
    for (int i = 0; i < length; i++) {

        // check if symbol is operator
        if (isOperator(post_exp[i])) {

            // pop two operands from stack
            string op1 = s.top();
            s.pop();
            string op2 = s.top();
            s.pop();

            // concat the operands and operator
            string temp = post_exp[i] + op2 + op1;

            // Push string temp back to stack
            s.push(temp);
        }

        // if symbol is an operand
        else {

            // push the operand to the stack
            s.push(string(1, post_exp[i]));
        }
    }

    // stack[0] contains the Prefix expression
    return s.top();
}

// Driver Code
int main()
{
    string post_exp = "ABC/-AK/L-*";
    cout << "Prefix : " << postToPre(post_exp);
    return 0;
}
```